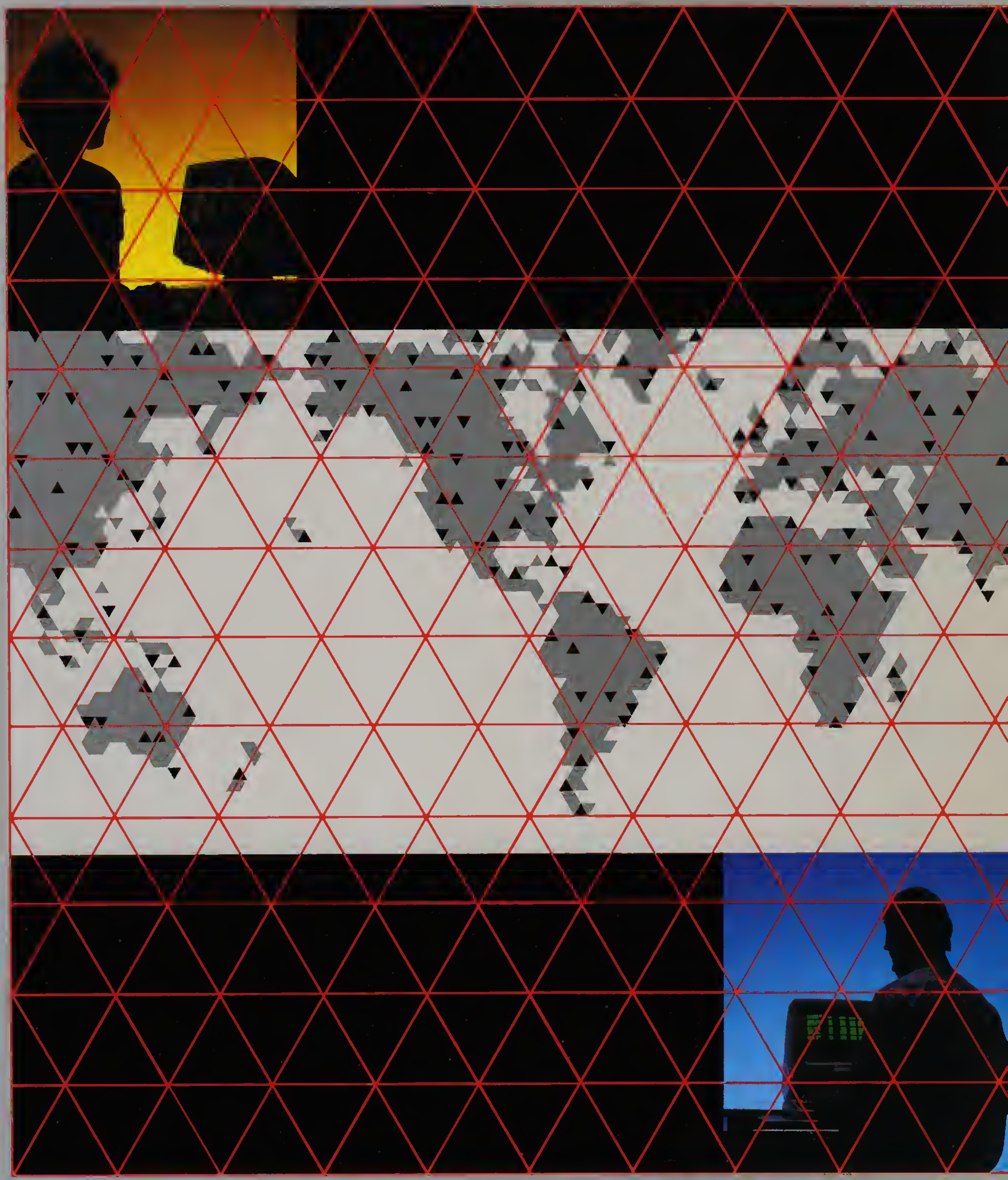
CDCNET Network Operations and Analysis



60461520

CONTROL DATA

CDCNET

Network Operations and Analysis Manual

Usage

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60461520

Manual History

Revision	System Version/ PSR Level	Date
A	1.0/647	December 1985
B	1.1/664	September 1986
C	1.2/678	April 1987
D	1.2.5/688	Septemper 1987
E	1.3/700	April 1988
F	1.4/713	December 1988
G	1.5.1/739	December 1989
H	1.5.2/750	June 1990
J	1.5.3/765	January 1991

This manual is revision J, printed in January 1991. It reflects CDCNET version 1.5.3 at PSR level 765, for operation on NOS version 2.7.1 at PSR level 750 and NOS/VE version 1.5.3 at PSR level 765.

This manual includes references to the CDCNET Network Management Station and changes to the Remote Line Monitor screens.

Also, various technical and editorial changes were made.

Contents

Figures

Tables

About This Manual

This manual describes the functions, procedures and commands associated with network operations of a CONTROL DATA® Distributed Communications Network (CDCNET). It presents CDCNET network operations concepts and guides you through the first steps of network operations.

Audience

This manual is written for the person needing information about CDCNET network operations activities and how to initiate them. The reader should have knowledge of NOS/VE and/or NOS concepts and operations, as well as an understanding of CDCNET's general purposes and concepts, as described in the CDCNET Conceptual Overview.

NOTE

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual. The CDCNET Network Management Station has utilities similar to NETOU and NPA. The CDCNET Network Management Station does not have a Dump Analyzer. Because of this, the chapters describing NETOU, Dump Analyzer, and NPA in this manual do not apply to CDCNET Network Management Station.

Organization

The following chapters are contained in this manual.

Chapter 1 gives you an overview of CDCNET from a network operator's perspective. You will learn about your role in the network, concepts important to you as a network operator, as well as the kinds of activities you may perform during operations.

Chapter 2 describes the Network Operator Utility (NETOU), which you use to monitor, control, and dynamically reconfigure CDCNET. NETOU is described for both NOS/VE and NOS environments.

Chapter 3 describes the NETOU session control procedures.

Chapter 4 is NETOU network control procedures.

Chapter 5 describes the Network Performance Analayzer (NPA).

Chapter 6 describes the reports and report formats generated by NPA.

Chapter 7 describes how to create customized NPA reports using IPF2 database files.

Chapter 8 describes the Device Interface Dump Analyzer.

Chapter 9 describes how to use NETOU, NPA, and the Dump Analyzer to analyze a network.

Chapter 10 describes how to use the Remote Line Monitor.

The appendixes include additional information to aid in understanding CDCNET.

- Appendix A contains a glossary of terms.
- Appendix B contains the ASCII character set.
- Appendix C contains the DI reset codes.
- Appendix D contains the procedures to enhance operator environment.
- Appendix E contains the Dump Analyzer MPB memory map.
- Appendix F contains the Dump Analyzer system tables.
- Appendix G contains the Dump Analyzer line and terminal control blocks.
- Appendix H contains the Dump Analyzer task and queue control blocks.
- Appendix I contains the Dump Analyzer error messages.

Conventions

The terms logic board and board are used interchangeably in this manual. They refer to any of the printed circuit board assemblies housed in the device interface (DI), such as the processor board, memory boards and line interface modules.

The terms Ethernet[1] and IEEE 802.3 are used interchangeably in CDCNET manuals. Ethernet refers to a network standard developed by Xerox, Intel, and DEC (Digital Equipment Corporation). IEEE 802.3 is the IEEE adaptation of that standard. The term IEEE 802.3 is a more precise label for the network standard. However, many network operations commands and software programs use the term Ethernet. CDCNET products covered by these standards are compatible with both IEEE 802.3 and Ethernet V.2.

The NOS 2 Operations and Analysis handbooks use the term COP (CDCNET Operator), which is the type of network operator described in this manual.

When descriptions and procedures apply to both a mainframe device interface (MDI) and a mainframe terminal interface (MTI), the term MDI is used for both device interface types. If it is necessary to specify both MDIs and MTIs in a section, they are specified in the initial instance, but from then on, only MDI is used.

All numbers in this manual are decimal (base 10) unless specifically identified as octal (base 8) or hexadecimal (base 16).

1. Ethernet is a registered trademark of the Xerox Corporation.

Related CDCNET Manuals

Manual Abstracts

Following is a brief description of each CDCNET manual.

Conceptual Overview	Discusses CDCNET in conceptual terms. It provides a broad view of CDCNET that explains the theoretical nature of this product. It does not attempt to define which particular product capabilities and features are currently available and which ones will follow in subsequent releases.
Product Descriptions	Provides reference, planning, and training information for customers who own or are interested in owning CDCNET products, and for Control Data personnel who use or work on CDCNET. The manual describes hardware and software products, provides information on how to select and use various types of network cables, and provides network configuration examples.
Terminal Interface	This is the primary manual for end-users who use interactive terminals to access computer services connected to CDCNET. The manual explains general terminal interface concepts, terminal commands and attributes, and connection attributes. For the advanced user, site administrator, and network analysts it also covers more advanced topics such as virtual and transparent modes, resolving communications problems, and the various terminal protocols supported by CDCNET.
Access Guide	This online manual guides the novice user through the process of accessing and using computer services through CDCNET. It includes procedures for connecting, disconnecting, and managing connections; displaying and changing terminal attributes; and terminal user exception processing. The more experienced user can find additional related information in the CDCNET Terminal Interface manual.
TCP/IP Programming Interfaces and Applications	Describes how to access the utilities that implement the TCP/IP protocols through CDCNET. The manual assumes the user is familiar with CDCNET terminal and connection attributes; knows the service title to access; and has some working knowledge and understanding of TCP/IP protocols.
Batch Device User Guide	Describes how to operate batch devices connected to CDCNET. It assumes the user is familiar with NOS and/or NOS/VE operating systems and with CDCNET access to these operating systems. The manual defines the concepts of I/O stations and provides the procedures for defining and controlling these stations. The online manual is available with NOS/VE and NOS operating systems.

Hardware Installation and Troubleshooting	Contains hardware installation procedures and troubleshooting guidelines for CDCNET hardware products and associated I/O cables. The manual is intended for individuals who install and check out CDCNET hardware products, operate them, add options to them, and maintain them.
Configuration Guide	Documents how to configure CDCNET software after it is installed on an operating system, and describes the responsibilities of the CDCNET network administrator. This manual also documents the Manage CDCNET Configuration Utility (MANCC), a utility for creating and editing files defining a CDCNET network.
DI Dump Analyzer	This manual is an online version of the DI Dump Analyzer section of the CDCNET Network Operations and Analysis manual. The manual is for CDCNET analysts who are familiar with Control Data host computer operating system concepts and operations. The manual describes how to use information from the Analyze CDCNET Dump (ANACD) utility to help troubleshoot network problems. Available with NOS/VE only.
Network Operations and Analysis	This manual documents how to monitor, control, and reconfigure CDCNET using the CDCNET Network Operator Utility (NETOU). The Network Operations section walks an operator through operations concepts, basic and advanced operations activities, and elementary troubleshooting decisions. The Network Analysis section describes the tools and methods used to analyze CDCNET performance including: instructions for using the CDCNET DI Dump Analyzer, a list of DI reset codes, a map of fixed address memory, and definitions of important system data structures. The NPA section of the manual provides information on how to generate various types of NPA reports and provides examples and descriptions of all NPA reports.

Diagnostic Messages	This manual is for network operators, network analysts, and programmers. The manual provides sorted lists of diagnostic messages and command responses issued by the CDCNET software. The primary sorted list of diagnostic messages describes the event causing each message and the appropriate user action. The primary sorted list of command responses describes the event causing the command response. Secondary sorted lists of diagnostic messages and command responses provide a cross reference of diagnostic message number and command response number to the CDCNET software products that issue the messages or command responses. The printed version of this manual is no longer available. However, a copy of the messages file can be printed on site. Available with both NOS/VE and NOS operating systems.
Commands	This manual contains all of the CDCNET Operator/Analyst commands. This manual is intended for operators, systems analysts, support engineers, and other experienced users.
CDCNET Network Management Station	This manual documents how to install, configure, and operate the CDCNET Network Management Station. The manual is for CDCNET operators and aministrators having previous experience as a UNIX system administrator.

Manual History

Not all sites find it convenient or expedient to install each new version and PSR level of CDCNET software. This presents a problem in maintaining sets of manuals that reflect installed software when later versions of CDCNET software are available but not installed. The following CDCNET Manual History table helps users to assemble and maintain the appropriate documentation by indicating which manual revisions support each release of CDCNET.

Manual/Audience Matrix

The CDCNET Manual/Audience matrix helps site planners, administrators, and users to determine their CDCNET documentation needs. The matrix categorizes each manual according to its type: overview, reference, tutorial, and so on. It then defines the audience of each manual in general terms: customer, end-user, LAN installer, and so on. Sites may have different audience designations for their audience, or may combine user functions.

CDCNET MANUAL HISTORY
RELEASE 1.3 - 1.5.3

CDCNET MANUALS	CDCNET RELEASE DATE/VERSION/PSR LEVEL					
	APR. '88 R1.3 L700	DEC. '88 R1.4 L716	JUN. '89 R1.4.2 L727	DEC. '89 R1.5.1 L739	JUN. '90 R1.5.2 L750	JAN. '91 R1.5.3 L765
	MANUAL REVISION					
CONCEPTUAL OVERVIEW 60461540	-	-	-	-	C	-
PRODUCT DESCRIPTIONS 60460590	B	C	-	D	-	E
TERMINAL INTERFACE 60463850	D	E	-	F	G	-
ACCESS GUIDE (ONLINE NOS) CDCNETA	X	X	-	X	-	-
ACCESS GUIDE (ONLINE NOS/VE) CDCNET_ACCESS	X	X	-	X	X	-
TCP/IP APPLICATIONS 60000214	B	C	D	E	F	G
BATCH DEVICE USER GUIDE 60463863	D	E	-	F	-	G
BATCH DEVICE USER GUIDE (ONLINE NOS/VE) CDCNET_BATCH	X	X	-	X	-	X
BATCH DEVICE USER GUIDE (ONLINE NOS) CDCNETB	X	X	-	X	-	-
HARDWARE INSTALLATION AND TROUBLESHOOTING 60000348	*	A	-	B	-	-
CONFIGURATION GUIDE 60461550	E	F	-	G	H	J
DI DUMP ANALYZER (ONLINE NOS/VE) ANACD		A	-	B	-	-
NETWORK OPERATIONS AND ANALYSIS 60461520	E	F	-	G	H	J
DIAGNOSTICS MESSAGES (ONLINE NOS) CNETMSG	X	X	X	X	X	X
DIAGNOSTICS MESSAGES (ONLINE NOS/VE) CDCNET_MSGS	X	X	X	X	X	X
CDCNET COMMANDS 60000414				A	B	C
CDCNET NETWORK MANAGEMENT STATION 60000568						A

M05352

NOTES:

- \- MANUAL NOT AFFECTED BY THE RELEASE.
- * RELEASES SUPPORTED BY LAN INSTALLATION MANUAL, DI INSTALLATION AND CHECKOUT MANUAL, AND TROUBLESHOOTING GUIDE.
- X INDICATES ONLINE MANUAL WAS UPDATED FOR THAT RELEASE.
- SHADED BOXES INDICATE THE LATEST REVISION LEVEL FOR THE MANUAL.

CDCNET MANUAL/AUDIENCE MATRIX

CDCNET MANUALS	MANUAL TYPE	AUDIENCE: Customer	End-User	LAN Installer	Customer Engineer	Network Operator	CE Support Engineer	Network Analyst	Site Administrator	Programmer
Conceptual Overview	Overview	■			■		■	■	■	■
Product Descriptions	Reference	■			■		■	■	■	■
Terminal Interface	User Guide		■			■	■	■		■
Access Guide	User Guide		■				■	■		
TCP/IP Programming Interface and Applications	Reference		■		■		■	■	■	■
Batch Device User Guide	User Guide		■				■	■		
Hardware Installation And Troubleshooting	Maintenance	■		■	■	■	■	■		
Configuration Guide	Ref./Tutorial					■	■	■	■	
DI Dump Analyzer	Ref./Tutorial				■	■	■	■		
Commands	Reference				■	■	■	■	■	■
Network Operations And Analysis	Ref./Tutorial					■	■	■	■	
Diagnostics Messages	Reference				■	■	■	■	■	■
CDCNET Network Management Station	Ref./Tutorial				■	■	■	■	■	

M05353

Additional Related Manuals

The following manuals contain helpful information.

Manual	Publication Number
Common Maintenance Software Interface	60455980
Concurrent Maintenance Library for Virtual Environment	60000019
Network Access Method (NAM) Network Definition Language (NDL)	60480000
NOS/VE System Usage	60464014
NOS/VE Commands and Functions	60464018
NOS Version 2 Reference Set, Volume 3	60459680
NOS Version 2 Reference Set, Volume 4	60459690
NOS Version 2 Analysis Handbook	60459300
NOS/VE System Performance and Maintenance, Volume 1	60463915
NOS/VE Network Management	60463916
Remote Batch Facility Reference Manual	60499600
IPF2 Reference Manual	84001950
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Operations Handbook	60459310

Ordering Manuals

Control Data manuals are available through Control Data Sales Offices or from:

Control Data
Literature and Distribution Services ARHLDS
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

You can also call (612)482-3800 or (612)482-3801, or FAX your enquiry to (612)482-3813. (If you are a Control Data employee, use the Controlnet number 235-3800, 235-3801, or 235-3813.)

Submitting Comments

Control Data welcomes your comments about this manual. Your comments may include your opinion of the usefulness of this manual, your suggestions for specific improvements, and the reporting of any errors you have found.

You can submit your comments on the comment sheet on the last page of this manual. If the comment sheet has already been used, you can mail your comments to:

Control Data
Technical Publications ARH219
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

You can also submit your comments through SOLVER, an on-line facility for reporting problems. To submit a documentation comment through SOLVER, do the following:

1. Select `Report a new problem or change in existing PSR` from the main SOLVER menu.

2. Respond to the prompts for site-specific information.

3. Select `Write a comment about a manual` from the new menu.

4. Respond to the prompts.

Please indicate whether or not you would like a written response.

Central Software Support Hotline

Control Data's Central Software Suppport maintains a hotline to assist you if you have trouble using our products. If you need help not provided in the documentation, or find the product does not perform as described, call us at one of the following numbers. A support analyst will work with you.

From the USA and Canada: (800) 345-6628

From other countries: (612) 482-3434

Introduction to Network Operations and Analysis 1

Introduction to Network Operations and Analysis 1

This chapter explains CDCNET concepts you should know before performing network operations. It also provides an overview of the CDCNET network operations processes. The following main topics are included in this chapter.

- Network Operations Concepts
- Network Operator Utility (NETOU)
- Session Control
- Network Control
- Network Performance Analyzer (NPA)
- NPA Reports and Report Formats
- How To Create Customized NPA Reports Using IPF2 Database Files
- Dump Analyzer
- Network Analysis
- Remote Line Monitor

It is important that you read this chapter before logging into CDCNET for network operations. For more information about CDCNET software and hardware concepts and terminology, review the CDCNET Conceptual Overview and the Product Descriptions manuals.

NOTE

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual. The CDCNET Network Management Station has utilities similar to NETOU and NPA. The CDCNET Network Management Station does not have a Dump Analyzer. Because of this, the chapters describing NETOU, Dump Analyzer, and NPA in this manual do not apply to CDCNET Network Management Station.

Network Operations Concepts

CDCNET is a distributed data communications network and a collection of data communications equipment interconnected by communications channels. CDCNET distributes its automated communications control and network management functions throughout the network, using a collection of device interfaces (DIs). DIs are connected to mainframes, terminals and printers, batch input and output equipment, and other networks. DIs may be connected to communications media that carry information formatted for CDCNET from one DI to another.

CDCNET may have a variety of configurations, depending upon the size of the network, number of terminals the network supports, and the amount of communications traffic the network generates.

The following are concepts you should read and understand before performing network operations.

Operating Environment

For CDCNET to operate, it must have:

- At least one host computer.
- For CYBER systems, one mainframe device interface (MDI) or one integrated communications adapter (ICA) is required. For CDCNET Network Management stations, an MDI or an ICA is not required.
- One terminal device interface (TDI) or mainframe terminal interface (MTI).

Host Computer

A host computer consists of a mainframe computer and its operating system. Together, they provide applications and services to the computer network.

The host computer can be:

- A Control Data Network Operating System/Virtual Environment (NOS/VE) mainframe
- A Control Data Network Operating System (NOS) mainframe
- A CDCNET Network Management Station (CNMS) on a UNIX system

The host computer is the network host. As a host, it can download software to DIs; provide programs to configure the network; and run other utilities needed by CDCNET, such as the utility that analyzes the CDCNET log file.

Device Interface

A DI is the main hardware device used to implement CDCNET. The DI controls access to the network and controls data communications through the network. Both DI hardware and software are modular. The type of hardware and software housed in a DI depends on the DI's specific function as a network communications controller. For more information about DI hardware and software, see the CDCNET Conceptual Overview and Product Descriptions manuals.

Lines, Trunks, and Network Solutions

You control several types of network communications media in CDCNET network operations; they are described in the following section.

Communication Line

A communication line connects data terminating equipment (DTE), such as a terminal or printer, to a DI. Data carried on this line from a DI is meant specifically for the terminal device, or is sent from the terminal device to the DI to which it is connected. Unlike a network solution, a line does not receive data meant for other areas of the network.

The DI hardware controlling communication lines includes the Communications Interface Module (CIM), Line Interface Modules (LIMs), and Unit Record Interface (URI) LIMs. The DI software that controls communication lines and the input to, and output from terminal devices, is called a terminal interface program (TIP).

Trunks

A trunk carries data for many devices connected to the network that may or may not be attached to the trunk. A trunk may be the underlying medium for a network solution. A trunk may also be the medium used to connect to a Public Data Network (PDN) through a gateway that acts as a translator between different protocols. The physical device for a trunk may be an Ethernet coaxial cable, a NOS/VE or NOS host's mainframe channel, a high-level data link control (HDLC) line, or an X.25 communication line.

Network Solution

Network solutions interconnect two or more CDCNET DIs or CYBER 93x hosts, using CDCNET protocols. A network solution is a trunk configured to carry both user data and CDCNET network management traffic.

A network solution is the main structural element of a CDCNET-type network. It can carry data from any point in the CDCNET network to any other point in the network. Unlike trunks and lines, it can also carry CDCNET network management services (such as log messages and alarms) and other services provided by the network (such as connections to host services).

Catenet

A catenet is a a set of one or more types of interconnected network communication media that use CDCNET protocols. The media that can be used to connect equipment together into a single catenet include the following: Ethernet local area networks (LANs), HDLC trunks, and X.25 network solutions. This term is often used in commands and text when referring to all the DIs and network solutions in a site's network.

Terminal Interface Programs (TIPs)

A TIP is a program that acts as a protocol translator between a terminal and CDCNET. CDCNET provides TIPs to support different terminal protocols. The following TIPs are provided by CDCNET in this release:

- Asynchronous TIP
- Telnet TIP
- Houston Automatic Spooling Protocol (HASP) TIP
- Unit Record Interface (URI) TIP
- 3270 Bisynchronous (BSC3270) TIP
- 3270 SNA Communications Protocol
- X.25 Asynchronous TIP
- X.PC TIP
- Mode 4 TIP
- Network Job Entry (NJEF) TIP
- Network Transfer Facility (NTF) TIP

These TIPs are defined by commands in DI system configuration procedures.

Logging Group

A logging group is a subset of DIs, within a catenet, that send their messages to a common log file. A logging group is established at configuration time. Each DI belongs to only one logging group. At configuration time, you can assign each DI the name of the logging group to which the DI belongs. The default logging group name on the configuration commands is CATENET. You can also configure each DI with the list of message numbers identifying the log messages it sends to the log file. Enable the default set of log messages by entering the DEFINE_SOURCE_LOG_GROUP command without the message parameter.

Operations Station

This manual uses the term operations station to refer to the remote terminal or host console from which operations activities are performed through NETOU.

Network Operations Activities

As a network operator, you control the network by managing the network's DIs and other network components, such as network solutions, communication lines, gateways, and by monitoring and responding to alarms and other messages generated by the network. These activities are performed by sending commands to DIs and observing the command responses.

You can monitor, control, and occasionally change the logical configuration of CDCNET either from an interactive terminal, or from a host computer console. Network operations commands are equivalent whether you perform network operations from an interactive terminal or host console.

The network activities you perform may vary depending on your site's configuration and communication needs. You may perform some activities more often than others, again depending on your site.

Gateways

A gateway is a program which connects two networks that use different protocols. CDCNET provides gateways to support translation between CDCNET and NOS protocols, CDCNET and X.25 network protocols, and CDCNET and Transmission Control Protocol/Internet Protocol (TCP/IP).

Network Products Gateways (NOS Only)

NOS supports a network based on 2550 Network Processing Units. This network has been known by various names, including Network Products and Network Host Products. The Network Products gateways allow information to be transferred between CDCNET and a non-CDNA NOS host. The Network Products protocol is different from the CDNA protocol; a gateway is necessary for CDCNET to access the NOS host.

Each NOS host uses an MDI or MTI to interface to CDCNET. An MDI or MTI provides the Network Products gateway function. Network Products connections exist between the gateway function in each MDI or MTI and its associated host. MDIs and MTIs containing Network Products gateways are members of both networks and understand both CDCNET and Network Products protocols. To CDCNET, a gateway is seen as the end of the connection, although a host mainframe is beyond the gateway.

There are two kinds of Network Products gateways: The terminal-to-application (T-to-A) gateway and application-to-application (A-to-A) gateway. The T-to-A gateway is called the Network Products terminal gateway (abbreviated as NP_TERMINAL_GW in network commands). The NP terminal gateway allows both interactive and remote batch terminal users to connect to the NOS host through CDCNET. There are two parts to the NP terminal gateway: The Interactive Virtual Terminal gateway (IVT gateway) and the Remote Batch Facility gateway (RBF gateway). The batch gateway depends on the interactive gateway. The NP terminal gateway software resides in a MDI or MTI. This gateway is an important portion of DI software. If the gateway is logically deleted or if the gateway software is removed from a DI, terminal users cannot connect to a NOS system.

The NP A-to-A gateway (abbreviated as NP_GW in network commands) is a gateway that allows applications on another NOS/VE, NOS, or foreign system to access the NOS system. The NP A-to-A gateway also allows applications on the NOS system to access applications on other NOS/VE, NOS, or foreign systems. File transfer (PTF) and job transfer (QTF) are the primary users of the NP A-to-A gateway.

X.25 Gateway

X.25 circuits allow CDCNET to access public data networks. An X.25 gateway is used to transfer data from a host connected to CDCNET, to a host in another network at the other end of the X.25 circuit. The X.25 gateway allows A-to-A connections to take place over an X.25 circuit. Some network commands control an X.25 gateway and can be used to start and stop access to X.25 services.

TCP/IP Gateway

The TCP/IP gateway supports CDCNET access to Department of Defense (DOD) networks and provides A-to-A services such as FTP. The gateway supports CDCNET access to Defense Data Networks (DDN) or workstations using TCP/IP protocols that support the Advanced Research Project's Agency Network (ARPANET) community. The gateway also supports the Excelan PC and equivalent products. There are network commands to control a TCP/IP gateway and to stop and start TCP/IP services.

Outcall Gateway

The CDCNET outcall gateway provides both terminal passthrough and device outcall services. This gateway must be present in the device interface before either service can be made available.

Terminal Passthrough

Terminal passthrough allows an asynchronous interactive terminal user on one line to establish a connection to an asynchronous device on another line. The device on the other line could be a terminal, modem, microcomputer, or non-CDCNET host such as NOS/BE, VAX, or IBM. With terminal passthrough, terminal traffic passes through the CDCNET network transparently and the two devices interface to each other as if they were directly connected. Terminal passthrough also supports connections to a modem with dial-out capability. See the Configuration Guide for detailed information on terminal passthrough.

Device Outcall

Device outcall allows host applications to initiate connections to asynchronous terminal devices. Once a connection is established, communications proceed as though the terminal had initiated the connection. Desktop/VE is the only NOS/VE application currently using device outcall. See the Configuration Guide for detailed information on device outcall.

Sending Network Commands

The following section describes concepts used to send commands to the appropriate destination in CDCNET.

Network commands must be sent to the network's DIs, affecting DIs and their hardware and software components. For example, there are network commands which display the operational status of a DI's logic boards, control statistics collection, add or delete lines from a network's configuration, stop communications on a network component, or run diagnostics on DI boards and ports.

To send a CDCNET network operations command to a DI, insert the command within another command which acts in the manner of an addressed envelope. This command is called SEND_COMMAND. It sends the network command to a specific DI or list of DIs. Session control commands are not sent to DIs to control network equipment, therefore they do not have to be sent within a SEND_COMMAND.

Command Responses and Alarms

In CDCNET, once a network command arrives at the proper destination, it is processed, and a response to the command is sent back to you.

Some messages are sent to you unsolicited, that is, without sending a command. These unsolicited messages are called alarms. Alarms are messages generated by network software for various events worthy of operator notification which the software detects, or for actions the software takes.

Physical and Logical Names

When sending network operations commands to ICAs or DIs, you can address DI components (boards, lines, trunks, network solutions, terminal devices) by name. The following naming conventions are allowed.

- Its physical name is derived from the physical addressing of a hardware component.
- Its logical name is derived from the value provided in the logical configuration of a DI. You can assign the logical name, which is the name by which the network's components and services are referenced.

Physical Names

Physical names are assigned to a DI's hardware devices, such as boards, ports, memory banks, terminal devices, communication lines, network solutions, and the DI itself. With the exception of boards, physical names are used as the default logical names for many DI components with logical names. Logical names are defined by CDCNET configuration commands. Once defined, the logical names are used in place of, and not in addition to, physical names. Some network operations commands, such as the online diagnostics commands, require that you specify physical names of devices.

Physical names begin with a $ character.

The physical name for a DI or ICA system is in the form

$DI_system_id

or

$ICA_system_id

where system_id represents the unique 12-character system ID assigned to the DI. An example of a DI physical name is $DI_0800253000A1.

For DI boards, the physical name is in the form

$devicen

The device portion of the name refers to board type, which may be one of the following values.

MPB	Main processor board
SMM	System main memory
PMM	Private memory module
CIM	Communications interface module
ESCI	Ethernet serial channel interface
ICA	Integrated Communications Adapter
MCI	Mainframe channel interface
LIM	Line interface module
URI	Unit record interface

SMM bank number (specified as BANK)

LIM port number (specified as PORT)

URI port number (specified as PORT)

The n portion of the name is a number that may have one of the following values.

- Board slot number (0 through 7). Refers to the board slot number of the hardware device in the DI. A DI contains two sizes of boards, large boards (MPB, PMM, SMM, CIM, ESCI, and MCI), and small boards (LIM/URI).
- System Main Memory (SMM) bank number (0 through 1).
- LIM port number (depending on the LIM model, either 0 or 1, 0 through 3, or 0 through 7). Port 0 is the top port on the LIM.
- URI port number (0 through 1). Note that only URI port 0 (the top port) is currently supported.

The following are examples of physical names for DI boards.

$CIM3	Physical name for CIM board in board slot 3
$ESCI4	Physical name for ESCI board in board slot 4

When a component is a subassembly of a device, such as a port on a LIM, the physical name of the subassembly is a concatenation of the main device name and the subassembly's name, joined by an underscore. For example, $LIM5_PORT2 is the physical name for the second port on a LIM board in LIM board slot 5, $SMM2_BANK0 is the physical name for bank 0 on a SMM board in board slot 2, and $URI7_PORT0 is the physical name for port 0 of a URI board in slot 7.

Figure 1oN-1 shows how physical names are assigned in a DI and shows an example of how boards may be installed in a DI. NN

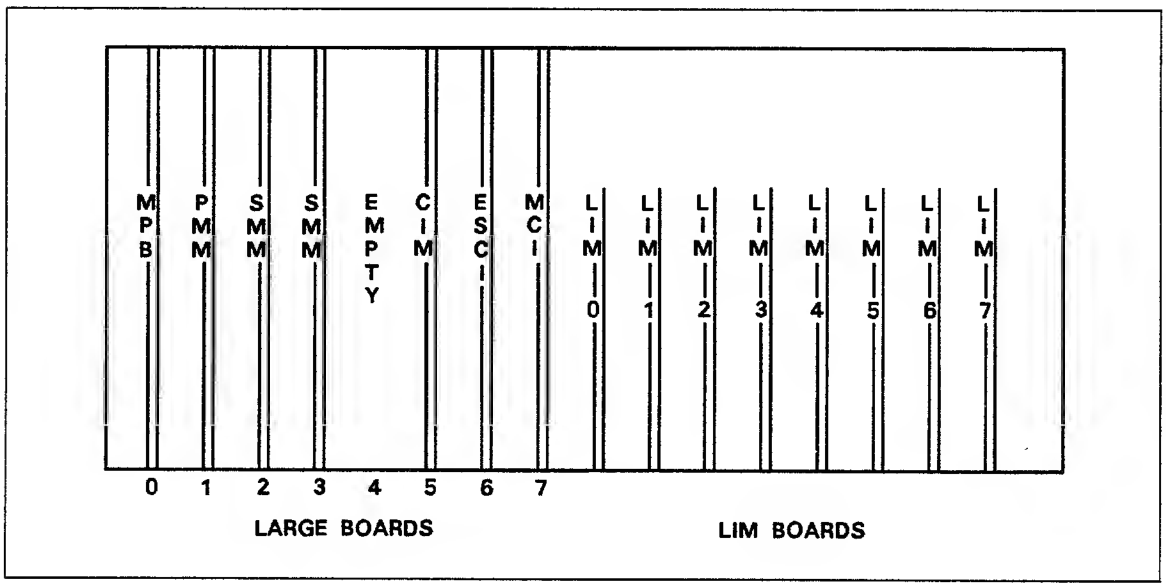


Figure 1-1. Example of Physical Names

Based on the configuration of boards shown in figure 1-1, the following physical names are assigned.

Large Board Physical Names	LIM Physical Names	Port Physical Names
$MPB0	$LIM0	$LIM0_PORT0 to $LIM0_PORT3
$PMM1	$LIM1	$LIM1_PORT0 to $LIM1_PORT3
$SMM2	$LIM2	$LIM2_PORT0 to $LIM2_PORT3
$SMM3	$LIM3	$LIM3_PORT0 to $LIM3_PORT3
$CIM5	$LIM4	$LIM4_PORT0 to $LIM4_PORT3
$ESCI6	$LIM5	$LIM5_PORT0 to $LIM5_PORT3
$MCI7	$LIM6	$LIM6_PORT0 to $LIM6_PORT3
	$LIM7	$LIM7_PORT0 to $LIM7_PORT3

Large board slot 4 is empty, therefore no physical name is assigned for slot 4.

A port's physical name is used as the default logical name for a communication line. For example, the default logical name for a line connected to LIM 0, port 0 on a DI is $LIM0_PORT0.

The physical name for a terminal device is made up of:

- $
- The type of terminal device.
- The last six digits of the 12-digit hexadecimal system ID to which the terminal is connected.
- The LIM number to which the terminal is connected.
- The port number which is connected to the communication line leading to the terminal device.
- The cluster address for the terminal device.
- The device address for the terminal device.

For example, given a terminal device configuration with the values:

System ID:	08002510003C
LIM number:	4
Port number:	2
Device type:	console
Cluster address:	00
Device address:	01

the terminal device would have the following physical name, which is the default logical name,

$CONSOLE_10003C_4200010000

Logical Names

Logical names allow you to give descriptive names other than physical names to network components, names which may be more immediately meaningful to your site than the physical names. For example, your site may choose to develop a descriptive naming scheme for communication lines. Defining short, descriptive logical names for DIs makes it easier for you to specify the system name when sending commands to DIs, rather than specifying the entire physical name. If logical names are not defined for components on configuration commands, the default logical name is the component's physical name. For example, if you do not define a logical name for a line, the line assumes a default name which is the physical name of the LIM and port to which the line is connected. If the line is connected to port 3 on LIM 1, then the default logical name for the line is the port's physical name: $LIM1_PORT3.

The CDCNET Configuration Guide contains conventions for creating logical names, and a table that shows the construction of logical names. The default logical names for network components are shown in the DEFINE command descriptions in the CDCNET Commands Reference manual.

A NETOU command, DISPLAY_LOGICAL_NAMES, displays the logical names defined for a DI, such as logical names for trunks, network solutions, and communication lines. See the command description in the CDCNET Commands Reference manual.

Example logical names:

Device interface names

North_Bldg_TDI_1

MDI_3C (for a DI with a system ID of 0800251003C)

TDI_134 (for a DI with a serial number of 134)

Trunk names

ESCI3

MCI2

Network names

Network_1

ESCI_Network

Line names

Engineering_Port_1

Line12 (for a line on $Lim1, Port 2)

Compsci_02

Addresses and Titles

Each DI has a unique address and title that identifies its location in the network. DI addresses are assigned during hardware installation. DI titles are assigned during software configuration. A configuration command (DEFINE_SYSTEM) may be used to define a logical name for the DI. The logical name maps to the DI's title and address. The system title is created from this logical name and from the default logical name. The difference between a title and other logical names known by a DI system is that titles are registered with a service called Directory Management Entity (ME) and may be known throughout the catenet; other logical names such as line names are local to the individual DI system. System titles are known throughout the catenet.

For example, suppose a DI is installed with the system ID of 0800252A1FF2 hexadecimal. This system ID is a part of its system address. During software configuration, the DI is defined as a TDI with the logical name First_Floor_TDI. The system title is then $SYSTEM_FIRST_FLOOR_TDI. You, as the system operator at configuration time, do not actually enter the portion of the system title represented by $SYSTEM_. The portion of the system title represented by $SYSTEM_ is the common prefix for all system titles assigned by convention. The NETOU generates this common prefix automatically. When operations commands are sent to this TDI, the logical name can be specified as the destination of the command and is interpreted as corresponding to the DI's system address and title, with the command received at the correct destination.

Network operations commands can also be sent to DIs by specifying their default logical names, which are described in the Physical Names section of this chapter.

It is important to keep track of titles, addresses and logical names. For suggestions on maintaining complete and accurate records of titles, addresses, and other network information, see the Recordkeeping section in chapter 4.

Network Configuration

The material in this section is intended to provide background information on the logical and physical configuration processes that ready CDCNET for operations. You do not have to understand the logical configuration process completely in order to perform the tasks described in this manual. Both the logical and physical configuration process should be completed by other site personnel by the time you begin CDCNET network operations. Defining and maintaining your site's initial logical configuration is the responsibility of the site administrator (the site administrator's responsibilities are documented in the CDCNET Configuration Guide).

CDCNET configuration involves planning and installing the network's hardware (physical configuration) and preparing the software used to run the network (logical configuration). Both physical and logical configuration must be completed before CDCNET can be operational.

Physical Configuration

The CDCNET Hardware Installation and Troubleshooting manual explains how device interfaces (DIs) and other network hardware components, including LAN cables and components (transceivers, repeaters, and multiplexers), are installed. This phase of configuration involves planning the physical layout, installing cables and lines, installing boards in the DIs, connecting the DIs to the network communications media, and ensuring that all the required hardware is present.

Logical Configuration

Logical configuration involves planning and preparing the software which runs in the DIs. The logical configuration is a description of functions of the DI and components connected to it. This description is in the form of configuration commands that define characteristics for the software which runs in the DIs. For example, configuration commands can be used to define the logical names of DIs and trunks and network solutions, to declare the line speeds for communication lines, and to define characteristics of batch devices such as printers and card readers. Configuration commands can also be used to define logical names for network components, such as DIs, trunks, network solutions, communication lines and terminal devices.

Logical configuration is necessary because DIs cannot function if they do not contain the software necessary to perform network tasks and operations.

The CDCNET Configuration Guide describes logical configuration. Logical configuration is the responsibility of a CDCNET site administrator, and should be accomplished prior to your beginning network operations. Occasionally during network operations, you may be directed to change the logical configuration while the network is running. For more information, see Changing Network Logical Configurations in the CDCNET Configuration Guide.

Network Validation

Network Validation is a feature that provides system security by requiring users to enter a valid username and password to use CDCNET. This username and password is in addition to login requirements for NOS/VE or other hosts. Network Validation is configured on a DI-by-DI basis and you can add it to any line serviced by the Asynchronous TIP, X.25 Asynchronous TIP, or Telnet TIP.

On a line configured for Network Validation, the DI prompts for a username and password before processing any user commands or executing any terminal user procedures. Terminal support software in the DI compares the entered and encrypted password with the one stored for the username. If the two passwords match, the user is accepted and normal processing continues. If the passwords do not match or the user does not have a password, the DI returns an error response and prompts the user to try again. This cycle repeats until either the proper password is entered, the retry limit is reached, or the connection times out.

The DI obtains the valid encrypted password for a username from the network validation database. This database is kept by the NOS/VE host providing file service to the DI. A site can use the LOAD_FILE command to load a DI with the validation files for users expected for that DI. However, each time the database changes the DI must be reloaded with the updated version. The network validation database is created and maintained with the Administer Network Validations utility. See the NOS/VE Network Management manual for more information.

The Network Validation feature also stores information on how network resources are being used. A network administrator can obtain this information from log messages and NPA reports. See the Configuration Guide for information on configuring Network Validation in a DI.

NTF Remote System Configuration

The Network Transfer Facility (NTF) is an application providing a fully symmetric queued file transport facility between a NOS/VE host and another host in a geographically dispersed network. Support for NTF is similar to batch device support for NOS/VE and CDCNET. NTF support on NOS/VE is similar to NJEF on NOS.

NTF supports IBM's Network Job Entry (NJE) protocol and HASP multileaving protocol for communication between hosts. The NTF network can include any of the following hosts:

- Multiple CYBERs running NOS/VE with NTF
- Multiple CYBERs running NOS with RBF or Network Job Entry Facility (NJEF)
- Multiple CYBERs running NOS/BE with INTERCOM5
- Multiple IBM, VAX, or other vendors' systems which support NJE and/or HASP

See the Configuration Guide for detailed information on NTF.

Network Operator Utility (NETOU)

The Network Operator Utility (NETOU) supports the set of commands and features used to monitor, control, and logically reconfigure CDCNET. NETOU supports commands to control the network from your operations station. Operations commands can be divided into the following types:

- Session control
- Network control

Session Control

Session control involves setting up and controlling your operations session. Examples of session control include controlling which DIs send alarm messages to your operations station, and routing NETOU command responses to a file that serves as a record of the responses. These activities do not actually control or change the network. See the CDCNET Commands Reference manual for descriptions of the commands used for session control.

Network Control

Network control activities include monitoring, controlling, and dynamically changing the logical definition of network equipment. See the CDCNET Commands Reference manual for descriptions of the commands used for network control.

Network Delay Measurement

Network delay measurement measures the average network delay time against a user-specified delay-time threshold. This allows you to evaluate the performance against a response threshold and report an error condition. See chapter 4 of this manual for detailed information on network delay measurement.

Network Performance Analyzer

The CDCNET Network Performance Analyzer (NPA) is a network analysis tool made up of flexible modular software components resident in a Control Data host computer. NPA helps you analyze the performance of your network by producing a variety of reports. These reports allow you to:

- Identify the configuration of your network
- Identify actual and potential hardware and software failures
- Identify potential congestion on communication lines
- Determine if your network is performing correctly
- Evaluate network use

NPA Reports and Report Formats

You may generate NPA reports individually to reflect a specific aspect of the network's performance, or you may generate a group of reports that reflect an overall picture of the network. Use NOS or NOS/VE commands to choose which report or set of reports NPA produces and the time period that your reports cover.

How To Create Customized NPA Reports Using IPF2 Database Files

You may also create NPA reports that are tailor-made for your specific needs. The IPF2 database files provide a process to change standard NPA reports. See chapter 7 for specific information on how to create customized NPA reports.

Device Interface Dump Analyzer

The DI Dump Analyzer program resides on a Control Data host computer and runs under NOS/VE or NOS. It processes subcommands that extract and format the information collected when DI memory is written to a dump file. The Dump Analyzer helps troubleshoot CDCNET by identifying events that have caused its DIs to reset.

Dump Analyzer subcommands let you display information about the conditions that existed at the time of the reset, including:

- **Important data structures**
- **Contiguous memory**
- **Program call chains**
- **Task control information**

Network Analysis

Analyzing the network requires a thorough understanding of the network, its configuration, and the CDCNET software that runs in DIs and on the host computer. Some advanced activities use several different programs that are not a part of NETOU. Other advanced activities can have a major effect on the network's performance, such as shutting off a DI or changing the network's logical configuration. Because advanced operations activities can affect the network's performance, your site may choose to have an analyst perform them, or to have you perform the activities under an analyst's supervision. Advanced operations activities include starting and stopping a gateway, stopping a DI, making online network configuration changes, and loading and unloading CDCNET software.

When problems occur in the network (such as users being unexpectedly disconnected from host services), CDCNET network commands can be used as a first step in troubleshooting. Network commands can be used to gather information about a problem and to isolate failures. Depending on the situation, you may be able to fix the problem yourself, using the available operations commands, or you may have to refer the problem to an analyst or customer engineer (CE). See the CDCNET Hardware Installation and Troubleshooting manual for more information on troubleshooting procedures.

Remote Line Monitor

The NOS/VE Remote Line Monitor monitors all received and transmitted characters on a given LIM and port. The given LIM and port must be supported by standard CDCNET CIM Firmware. See chapter 10 of this manual for detailed information on the Remote Line Monitor.

Network Operator Utility (NETOU) 2

Network Operator Utility (NETOU) 2

This chapter describes the Network Operator Utility (NETOU), as used in NOS/VE and NOS environments. NETOU allows you to access CDCNET and perform network operations activities from a remote terminal or from a host console.

The command syntax is the same whether you are at an interactive terminal or host console. However, some aspects of terminal and console command entry, display and screen control are different. This chapter explains those differences.

Since NETOU has some different features on NOS/VE and NOS, the chapter is divided into three sections:

- Common Network Operations Features
- Operations in a NOS/VE Environment
- Operations in a NOS Environment

NOTE

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual. The CDCNET Network Management Station has a utility similar to NETOU.

Common Network Operations Features

This section describes features of NETOU that are common to both NOS/VE and NOS operations environments. The following features are described: Command syntax rules, descriptions of common command verbs, order of command execution, command responses, alarms, and severity levels for responses and alarms.

Command Syntax

This section outlines the syntax rules for the CDCNET commands described in this manual. All commands follow a subset of the NOS/VE SCL syntax. This section is provided to give you sufficient information to understand the commands used in this manual. For more information on SCL command syntax, see the NOS/VE Commands and Functions manual. Commands used in a NOS environment have additional properties (see Command Syntax for NOS NETOU later in this chapter).

Command Format

A command is in the following form.

```
command_name parameter_1=value_1,parameter_2=value_2,...
```

Example:

```
DISPLAY_HARDWARE_STATUS DEVICE_NAME=$LIMO_PORTO,DISPLAY_OPTION=EXPANDED
```

Either a blank or a comma can be used as a separator. The underscore character cannot be omitted. Command strings may be up to 256 characters long. The maximum size of a SEND_COMMAND command (SEND_COMMAND plus command to be sent) is 512 characters on NOS and 65K on NOS/VE. You may continue entering a command on another entry line using the ellipsis (..), as shown in the following example.

```
senc c='start_process_metrics p=commands,..
g=(summary,expanded)',s=mdi_3
```

Command Abbreviations

Command names are abbreviated by taking the first three characters from the verb portion of the command name and combining them with the first character from the remaining words in the command. The abbreviated form of a command name having a plural form is the same as the abbreviation for the singular form.

For example, DISPLAY_HARDWARE_STATUS is abbreviated by taking DIS from DISPLAY and combining it with the H from HARDWARE and the S from STATUS to form.

DISHS

Parameter Abbreviations

Parameter names are abbreviated by taking the first character from each word in the parameter name. For example, the parameter LINE_NAME has the abbreviated form LN.

Parameters and Parameter Values

Parameters consist of a parameter name followed by an equal sign and a parameter value. A parameter value may be a list of values, as in:

parameter=(value_1,value_2,...)

or a list of lists, as in:

parameter=((value_1,value_2),value_3,(value_4,value_5...)...)

The following types of parameter values are allowed: string, name, integer, boolean, and keyword value.

A string is any sequence of ASCII characters enclosed by apostrophes ('). Most of the network operations commands must be entered as a string value within SEND_COMMAND. The enclosed command string must be surrounded by apostrophes. If you include an apostrophe within a string value, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized, as in the following example.

```
send_command c='write_terminal_message,..
m=(''New communications configuration tomorrow'',''Network down ..
until 10:00.'')',s=tdi1
```

An SCL name is a combination of from 1 through 31 alphabetic characters (ASCII characters A through Z and a through z), digits (ASCII characters 0 through 9), and/or special characters (underline [_], dollar sign [$], number sign [#] and commercial at [@]). Lowercase is folded to uppercase in a name. The first character cannot be a digit.

An integer parameter value represents a binary, octal, decimal, or hexadecimal integer value. Integer values may be expressed as a combination of digits or for hexadecimal integers, A through F (uppercase or lowercase). A hexadecimal integer must begin with a digit. SCL makes no distinction between uppercase and lowercase characters in hexadecimal integer constants.

Integer parameter values may be expressed as: integer (radix), or a range of integer values. If you do not specify a radix, the decimal system (base 10) is assumed. Any radix between 2 and 16 is accepted. A radix must be surrounded by opening and closing parentheses, as in 1FFFF(16) and 101(8).

In command descriptions, when two integers are separated by an ellipsis (..), a range of integer values is possible. The allowed value may be the first value through the second value. For example, the parameter value BUFFER_SIZE = 64..4096 indicates that any value from 64 through 4096 is possible for the BUFFER_SIZE parameter.

A boolean parameter value represents a condition of either TRUE or FALSE. In NOS/VE systems, there are three possible words used for both TRUE and FALSE conditions. For a TRUE condition, you may specify TRUE, YES, or ON. For a FALSE condition, you may specify FALSE, NO, or OFF. In NOS systems, NETOU only supports YES and NO.

A keyword value is a parameter value that has a special meaning in the context of a particular parameter. For example, the command DEFINE_LINE has a parameter LINE_TYPE, where two types of lines, switched and dedicated, are allowed. Two keyword values are allowed for this parameter: SWITCHED and DEDICATED. You specify one or the other by providing the appropriate keyword value for the parameter. The keyword value ALL is frequently used in commands to select all available options for a parameter value.

Default Parameter Values

Not all parameters require you to provide values. In the command descriptions, required and optional parameters are designated. Most parameters have a value called a default parameter value that is provided if you do not specify the parameter with the command. Default parameter values are specified in command descriptions.

Command Entry

You can enter the parameters in this manual in two ways.

- Position-dependent
- Position-independent

In position-dependent format, you supply values for parameters in the order specified in the command format, without entering parameter names or equal signs. Separate parameter values with commas. If you omit any parameters, you must supply a comma for the missing parameter.

In position-independent format, you supply the values for the parameters by specifying the parameter name and the equal sign before the value for each parameter. You can enter the parameters in any order.

Command Verbs

This section explains the verbs used in several common network operations command types. Commands beginning with these words comprise the majority of network operations commands.

Add

Add commands add to the logical configuration of an element you specify. Add commands are a part of the set of configuration commands, and are used in DI configuration files.

Cancel

Cancel commands delete the logical configuration of the element you specify. For example, you may cancel the logical configuration of an Ethernet network solution using the CANCEL_ETHER_NET command. The network solution's logical configuration is deleted. If you want the network solution to support data transfer again, you must redefine the network using a DEFINE command type (see following descriptions).

Change

Change commands change the current logical configuration of a hardware component, the values of certain aspects of a DI's operating system such as buffer size and memory management, or the set-up of the network's system for reporting alarms.

Define

Define commands create a logical configuration of the element you specify in the network. Define commands are a part of the set of configuration commands, and are used in DI configuration files. These commands are also used if you cancel a component's logical configuration and want to redefine it.

Delete

Delete commands delete from the logical configuration of an element you specify. For example, you may delete an X.25 gateway outcall title, which was previously added by an ADD command, from the logical configuration of the X.25 gateway.

Display

Display commands return information you request to your operations terminal or console screen. There are display commands to display the following information.

- Status for hardware and software elements of a DI.
- Configuration parameters for network elements.
- The list of log messages and alarms to be transmitted from a DI.
- The current date and time registered at a specific DI.
- Diagnostic test results.

For commands that display several parameters, you can select which parameters you want displayed. These commands have a parameter called DISPLAY_OPTION (DO), which allows you to specify only parameters that are of interest to you. You may choose one, several, or all of the options that a DISPLAY_OPTION parameter allows.

For example, the DISPLAY_SYSTEM_OPTIONS (DISSO) command, which displays the current value of DI system program attributes, has a DISPLAY_OPTION parameter. DISPLAY_OPTION allows you to choose from among several configuration attributes you want displayed, by specifying keyword values such as DATA_BUFFER_SIZE, BUFFER_PERCENTAGE, MEMORY_MANAGER_PERIOD, and CLOCKING_SYSTEM.

Start

Start commands begin the specified action, or enable the specified component to begin data communications. Some start commands make an element you specify operational, or ready for data transfer. For example, you may start communications traffic on a communication line from a LIM to a terminal using the command START_LINE. Other start commands begin online diagnostic tests (such as START_CIM_TEST and START_ESCI_TEST), and statistics collection (such as START_LINE_METRICS and START_NETWORK_METRICS).

Stop

Stop commands end the specified action, or disable the specified component from performing data communications. Some stop commands stop the support of data transfer on the network element you specify, such as STOP_LINE and STOP_NETWORK. Other stop commands stop diagnostics, and statistics collection.

Order of Command Execution

Commands you send to a DI are executed in the order received. Commands from operators at different stations that affect overlapping sets of DIs may be received in a different order at each DI. If there is more than one CDCNET network operator currently logged in and sending commands, there is no guarantee that commands sent from one network operator to network components are performed in sequence before those sent from another network operator.

Command Responses

All commands entered generate a response. This section describes command responses.

Command Response Format

CDCNET command responses have the following format (brackets indicate optional portions of the response).

```
FROM tttttttttttttttttttttttttttttttt  ccccc
[--ssssssssssss-- ]response text
```

tttttttttt... Logical or physical name of system sending response.

ccccc Numerical identifier for the command response. NOS/VE does not display this identifier for informative command responses; NOS does. Using the message number, you can reference the command response's description in the online CDCNET Diagnostic Messages manual.

ssssssssss Severity level of command response (see Severity Levels for Command Responses and Alarms, later in this chapter) This severity level is not displayed for every response. If no severity level is displayed, the response is informative and the command has completed successfully.

response text The response text may either directly follow the severity level or begin on the next line.

For NOS/VE environments, a normal CDCNET command response is written to the output file when it includes response text. An abnormal CDCNET command response is always written to the standard file $RESPONSE.

The following is an example command entry and command response (NOS host). It shows a command called DISPLAY_HARDWARE_STATUS (DISHS) being sent to a DI, and the response sent back to the network operator.

Command:

```
senc s=mdi_1,c='display_hardware_status'
```

Command response:

```
FROM MDI_1                                      33021
Hardware Status
device name     status     state        version    lim/bank/port   type
$MPB0           on         active       0000
$PMM1           on         active       0008
$SMM2           on         active       0001       2
    3           off
$CIM4           on         configured   0001       0,1,2,3
$CIM5           down       not config.  0001
$ESCI6          on         active       0000
$MCI7           on         active       0000
$LIM0           on         not config.             4               RS232
$LIM1           down       configured              4               RS232
$LIM2           on         configured              2               RS449
$LIM3           on         not config.             2               RS449
```

The following is an example command entry and command response (NOS/VE host). It shows a command called DISPLAY_HARDWARE_STATUS (DISHS) being sent to a DI, and the response sent back to the network operator.

Command:

```
senc s=mdi_1,c='display_hardware_status'
```

Command response:

```
FROM MDI_1
Hardware Status
device name     status     state        version    lim/bank/port   type
$MPB0           on         active       0000
$PMM1           on         active       0008
$SMM2           on         active       0001       2
    3           off
$CIM4           on         configured   0001       0,1,2,3
$CIM5           down       not config.  0001
$ESCI6          on         active       0000
$MCI7           on         active       0000
$LIM0           on         not config.             4               RS232
$LIM1           down       configured              4               RS232
$LIM2           on         configured              2               RS449
$LIM3           on         not config.             2               RS449
```

Other examples (NOS host):

```
send_command c='display_date_and_time',s=di_sn093

FROM DI_SN093                                    33525
 System date and time
      31/01/85 23:20:24
```

Other examples (NOS/VE host):

```
send_command c='display_date_and_time',s=di_sn093

FROM DI_SN093
 System date and time
      31/01/85 23:20:24
```

Common Responses

The following command responses are common to all network commands.

- Responses indicating that the DI or component cannot be located or is unavailable may occur for any command sent to a DI.
- Error responses indicating unknown commands, invalid parameters, and incorrect parameter values (command parser errors which abort execution of commands).

These common responses are not documented with the commands responses. Only responses that are uniquely defined for the command are documented. All command responses are documented in the online CDCNET Diagnostic Messages manual.

Loss of Commands and Responses

Network commands to specific DIs are sent by transport connections that ensure commands are delivered to the correct DI and that loss of commands in transmission cannot occur. However, a destination DI could fail while the command is executing, or the command processor in the DI could stop abnormally. To allow for such events, NETOU times the response for any command and declares a command failed if no response is received from the CDCNET system within 120 seconds after the command is sent. For commands that do not send a response within 120 seconds, the following response is sent.

NOS example:

```
--ERROR-- No response received from system <name> for the CDCNET command
<command_name>.
```

NOS/VE example:

```
--ERROR--No response received from system <name> for the last CDCNET command
```

Break Processing (Response Suppression)

With break processing, you may suppress responses to network commands in progress (keep any output from commands from being displayed on your screen). Commands with suppressed responses complete, but no response for the commands are delivered to your operations station.

Command response suppression does not abort command processing. You cannot abort commands that are being processed at the destination DIs. Once received at a DI, commands complete regardless of what you enter from your terminal or the host console. When you suppress responses, the next command entry prompt (nou/ on NOS/VE and NOU/ on NOS) indicates the end of response suppression. Commands entered after you receive that prompt execute normally and return responses.

Response Suppression on NOS/VE

On NOS/VE, you initiate response suppression by entering the user_break_2 at an interactive terminal. If an included file is executing when response suppression is initiated (see Building Command Files, in chapter 3 of this manual) response suppression both suppresses responses for commands in progress and terminates NETOU processing of the file.

When a user break sequence is entered, NETOU responds with the Terminal Manager response to a user break. The response to the user break also identifies commands for which responses have not been received and commands that have unknown destinations. The following messages are used to indicate these conditions.

```
No response received from system <string> for the last CDCNET command.

System <string> is unknown.

No response received to connect request to system <string>.
```

Response Suppression on NOS

On NOS, when a break command is issued, some commands sent to a DI may still be processed, and others may have the output discarded. You initiate response suppression using one of the following methods.

- At an interactive terminal, enter the user_break_1 or user_break_2 sequence. NETOU responds with the following message.

  ```
  Pending responses suppressed
  ```

- At a host console, enter K./

You can enter a command response suppression command while a file of network operations commands is being executed (see Building Command Files in chapter 3 of this manual). Command response suppression both suppresses responses and terminates NETOU processing of the command file.

Alarms

Alarms may be sent from DIs to your operations station during an operations session. These alarms are unsolicited; they are not responses to commands, and you may receive them at any time during an active NETOU session.

On NOS, alarms are always activated. You do not have to enter a command to activate their transmittal to your operations station. In NOS/VE environments, alarms are not initially activated. You must explicitly activate alarms by entering the ACTIVATE_ALARMS command before you can receive alarms. Rather than manually activating alarms every time you begin an active NETOU session on NOS/VE, you can automatically activate alarms through your NETOU prolog by placing the appropriate commands in your prolog. See Session Control on NOS/VE in chapter 3 of this manual.

Alarms alert you to a wide range of conditions that occur in a network, from the completion of a diagnostic test to the failure of a hardware component. In addition, any messages sent to you from the network's terminal users appear as alarms at your display.

When a DI completes being loaded and logically configured, alarms generated during the logical configuration are sent to your operations station.

Much of your network operations work involves responding to CDCNET alarms.

Alarm Format

CDCNET alarms have the following format (brackets indicate optional portions of the response).

```
****** ALARM FROM  tttttttttttttttttttttttttt  date/time  ccccc
[--sssssssssssssssss-- ]alarm text
```

tttttttttt... Logical or physical name of system sending alarm. An alarm generated by NETOU itself, such as an alarm issued when an MDI connection is broken, displays NETWORK_OPERATOR_UTILITY in this field.

ccccc The numerical identifier for the alarm message. This identifier is displayed for all alarms and is intended to help you index into the online CDCNET Diagnostic Messages manual for a description of the message.

ssssssssssss Severity level of the alarm (see Severity Levels for Command Responses and Alarms, later in this chapter). This field is suppressed for informative alarms; if no severity level is displayed, the alarm is informative. Informative alarms may indicate the completion of an operation (such as a diagnostic), the recording of information (such as statistics), or convey a message from a terminal user. Informative alarms are not the result of incorrect or incomplete CDCNET operation.

The following is an example of an alarm:

```
****** ALARM FROM DI_SN093                  85/01/31  23.24.31  458
       New maximum recovery rate
       Failure ID = 0013
       Threshold count = 1
       Period in seconds = 2
```

Alarm Output

When alarms are sent to you, they are immediately displayed at your screen unless you specifically route alarms to a file only, using the ROUTE_ALARM command on NOS, or connect the alarm output to a file on NOS/VE.

Alarms appear in the order received. At an interactive terminal, alarms are not displayed while you are entering input. Should an alarm be delivered to your terminal, an alarm bell rings only at interactive terminals, and only on NOS NETOU. At an interactive terminal or host console, alarms may be interspersed within the responses to commands.

Severity Levels for Command Responses and Alarms

The command responses and alarms you receive are grouped into the following severity levels: Informative, Warning, Error, and Fatal.

Informative and Warning command responses indicate a command completed successfully. Error and Fatal are both error responses alerting you to command errors. The following paragraphs are descriptions of these severity levels.

Informative

An Informative command response indicates successful command completion. Informative alarms are not the result of incorrect or incomplete CDCNET operation. The severity level for informative responses and alarms is not displayed. If you receive a response or alarm without a severity level displayed, the response or alarm is informative.

Warning

A Warning command response indicates that a command completed successfully, but that the command may have unintended effects. For example, some of the definition parameters for a communications trunk may be changed while the trunk is active. Changing those parameters, however, could disrupt communications over the trunk, unless changes at both ends of the trunk are coordinated. Warning responses are sent for redundant commands.

Warning alarms alert you to potential network problems. They indicate that a DI or network is approaching an error or fatal condition, such as a lack of system buffers. However, no operation is yet incorrect or incomplete due to the condition. Check the alarm's text to determine what you can do to avoid errors in the network.

Error

An Error command response indicates that a command failed due to operator error. An error response may indicate, for example, errors detected in command processing, errors in parameters, such as unknown names, and attempts to execute a command which is not allowed. Error responses may also indicate that a connection could not be established to deliver a command to its destination system.

Error level alarms indicate the following: the failure of an operation to complete correctly, with the possibility of being recovered by the DI's software; and the failure of a device connected to the DI, such as the loss of a modem signal or communication line.

Fatal

A Fatal command response indicates that a command failed due to device failures or lack of resources to complete the command. For example, if there is not enough memory available on a DI hardware device to execute a command, a Fatal-severity level response would be returned.

Fatal alarms indicate the following: the failure of an operation to complete correctly, without the possibility of being recovered (such as the failure of DI system software); and the failure of tasks in the DI system software.

NOTE

When you receive fatal alarms, it is important to intervene when possible to prevent a system failure.

Operations in a NOS/VE Environment

Figure 2-1, NETOU Operating Environment for NOS/VE, shows the major software and hardware components that provide the operations environment on NOS/VE. For NOS/VE environments, NETOU consists of the CYBER-resident NETOU application, and the Dependent Command Management Entity resident in each DI. On NOS/VE, you log into a NOS/VE service title, and enter a command to invoke NETOU. Selecting NETOU allows you to add the subset of NETOU session and network control commands to the NOS/VE commands you are currently allowed to enter. You may continue to enter other NOS/VE commands during any active session with NETOU.

Accessing NETOU

Before accessing NETOU, you must connect to a service on the host system to begin an interactive terminal session. Use the CREATE_CONNECTION (CREC) and the service title defined at your site through the Manage_Network_Applications Host Utility. The following is an example of a connection to a service on NOS/VE entitled, NVE.

```
create_connection nve
```

If you need to review how to use the CREATE_CONNECTION command, see the Terminal Interface manual.

To access NETOU, first log in to NOS/VE using the standard NOS/VE login process.

You must be validated to use NETOU. Access to NETOU is controlled by the NOS/VE operating system. Your site's family administrator, through the Administer Validation Utility, controls the NETOU privileges available to you. Check with your site's administrator if you are not validated to use NETOU.

To use NETOU, enter the following NOS/VE command after you have logged in.

NETWORK_OPERATOR_UTILITY (NETOU or NOU)
PROLOG = file reference
STATUS = status variable

Both parameters are optional. The PROLOG parameter specifies a file containing commands to be executed once NETOU is invoked. Any NOS/VE or NETOU command can be in the prolog. The default file reference for the PROLOG parameter is $USER.NETWORK_OPERATOR_PROLOG. If you do not specify this parameter and the default prolog file does not exist, no prolog file processing occurs. For more information on prologs, see Creating a Prolog in chapter 3 of this manual. For information on the STATUS parameter, see the basic status concept for NOS/VE SCL in the NOS/VE System Usage manual.

You may add the NETOU command to your NOS/VE prolog (see the System Access chapter of the SCL for NOS/VE Commands and Functions manual), so that NETOU is automatically invoked each time you log in.

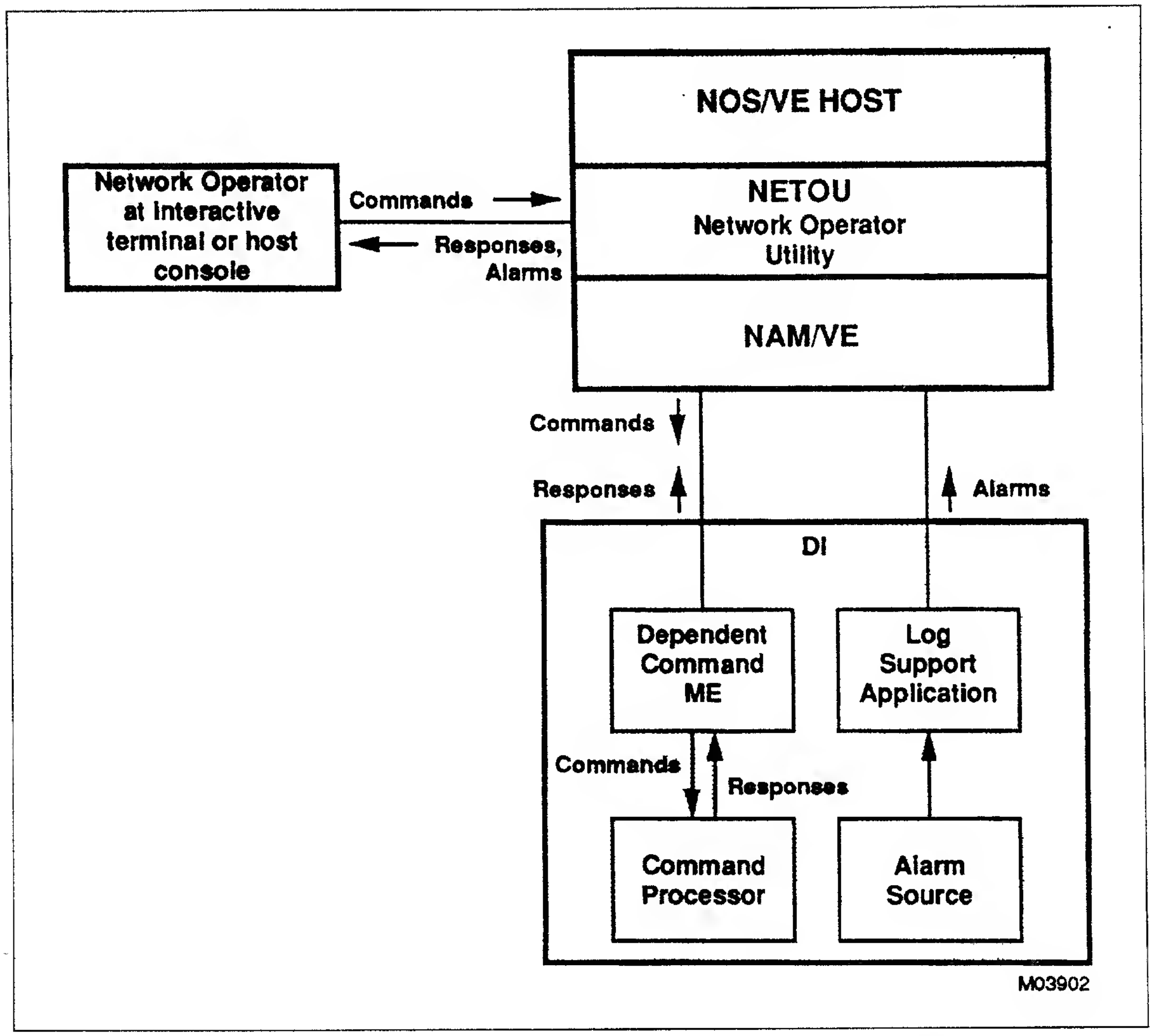


Figure 2-1. NETOU Operating Environment for NOS/VE

Prompts for NETOU

The prompt for NETOU is:

```
nou/
```

This prompt indicates that you have selected NETOU and can begin entering NETOU commands. You may also enter other NOS/VE commands.

You may also receive the following prompt for NETOU.

```
nou../
```

This prompt indicates that the previous line you entered was continued to another line.

Paging

Paging allows you to move forward within a display on the terminal screen. You may enable or disable paging using the CDCNET terminal command CHANGE_TERMINAL_ATTRIBUTES (CHATA). To do this, enter the network command character (NCC) (shown here as a percent sign [%], but the actual NCC may differ for your terminal), and the CHANGE_TERMINAL_ATTRIBUTES command, as in the example that follows. You may also enter the CHANGE_TERMINAL_ATTRIBUTES command without a preceding network command character to cause the host version of the command to execute.

```
%CHANGE_TERMINAL_ATTRIBUTES HOLD_PAGE=ON or OFF
```

ON enables paging; OFF disables paging. The default is for paging to be OFF. When paging is on, to scroll to the next page of text, enter a carriage return or a control character. For more information about the CHANGE_TERMINAL_ATTRIBUTES command and the network command character, see the Terminal Interface manual.

NETOU Terminal Display Format

NETOU at a terminal uses virtual line mode format (as opposed to full screen mode) for display output. Commands are entered and responses are returned line by line. You use some utilities to perform network operations tasks that use full screen mode. These utilities run outside of NETOU, and include the Network Performance Analyzer (NPA), and the Manage CDCNET Configuration Utility (MANCC).

Exiting NETOU

To exit NETOU, enter the QUIT command.

```
quit
```

When you enter QUIT, you can exit NETOU and still remain logged in to the service. QUIT removes the NETOU commands from the set of commands you are allowed to enter. The LOGOUT command both terminates NETOU and logs you out of Timesharing.

Entering Network Commands

NETOU commands are valid only within a NETOU session. The session begins when you enter the NETOU command to invoke NETOU. The session ends when you enter the QUIT command. You use SEND_COMMAND to send network commands to the appropriate destination.

SEND_COMMAND (NOS/VE Version)

A network command is embedded within SEND_COMMAND as a string value, and another parameter sets the destination for the network command. SEND_COMMAND has the following format on NOS/VE.

SEND_COMMAND
COMMAND = string
SYSTEM = list of name
OUTPUT = file name
STATUS = status_variable

There are two required parameters: COMMAND and SYSTEM. COMMAND is the CDCNET operations command to be sent to the specified DI. The command is entered as a string value enclosed by apostrophes (').

NOTE

If the command you are sending contains any apostrophes, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized. Otherwise, NETOU assumes the embedded apostrophe signals the end of the NETOU command, and errors could result.

For example, the following command contains an embedded apostrophe in the message being transmitted to all terminals connected to TDI1.

```
send_command c='write_terminal_message,..
m=''ENGINEERING''''s network down until 10:00''',..
s=tdi1
```

SYSTEM is the logical or physical DI name or list of DI names to which the command is to be sent. If a CDCNET command is sent to more than one CDCNET system, a response must be received from each system for the command to complete. The other parameters are optional.

SEND_COMMAND Example

The following command sequence would be entered to stop traffic on a communications line connected to a DI named TDI_3.

```
send_command command='stop_line line_name=line3',system=tdi_3
```

The actual command to stop communications traffic is enclosed within a SEND_ COMMAND command that specifies the DI (TDI_3) to which the line is connected.

Operations in a NOS Environment

Figure 2-2 shows the major software and hardware components that provide the operations environment on NOS. On NOS, NETOU is an application that you select as you would other NOS applications, such as Interactive Facility (IAF). For the NOS environment, NETOU consists of the CYBER-resident NETOU application; the Operator Support Application (also known as the Independent Command ME) which resides in MDIs/MTIs that have been chosen, during logical configuration, to provide operator support; and the Dependent Command ME which is resident in each DI in the network. When you select NETOU, your job is dedicated to NETOU until you exit that application. Commands other than NETOU session and network control commands are not accepted.

On NOS, NETOU can be used either at a remote terminal or a NOS host console. On the NOS host console, NETOU runs through the NAM K display. The K display has special character and command entry restrictions that are described in the section titled Network Operations from a NOS Host Console later in this chapter.

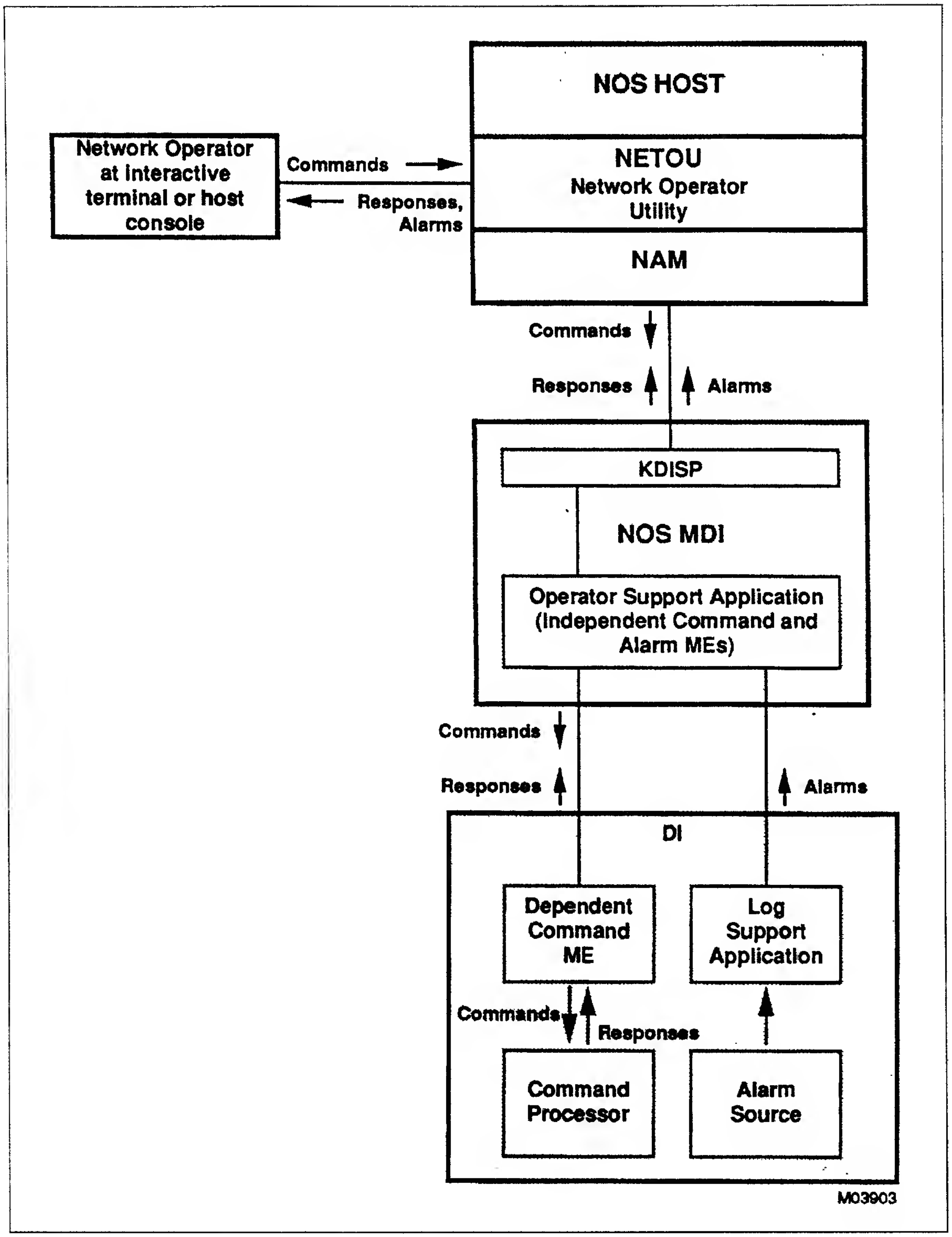


Figure 2-2. NETOU Operating Environment for NOS

Network Operations from an Interactive Terminal

At an interactive terminal, you communicate to CDCNET via NETOU through a normal interactive terminal connection.

NETOU Terminal Display Format

At a terminal, NETOU uses virtual line mode (as opposed to full screen mode) for display output. Commands are entered and responses are returned line by line. You use some utilities to perform network operations tasks that use full screen mode. These utilities run outside of NETOU, and include NPA, Network Logfile Termination Utility (NLTERM), and MANCC.

Login

Use the following procedure to log in to NOS and select NETOU from a host console.

1. Create a connection to the host system using the CREATE_CONNECTION (CREC) terminal user command, as in the following example in which the user creates a connection with the host system by specifying the title NOS100.

   If you need to review how to use the CREATE_CONNECTION command, see the Terminal Interface manual.

   ```
   create_connection,nos100
   ```

2. To enter NETOU, log into NOS and select the NETOU application. Enter your family, user name, password, and NETOU, separating each by commas (if you use the default family name, log in beginning with a comma).

   ```
   family,user name,password,netou
   ```

3. If you are validated to access NETOU, you are connected to NETOU and see the following message.

   ```
   WELCOME TO NETWORK OPERATOR UTILITY
   CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1989.
   ```

4. If you are not validated to access the NETOU application, you receive the following response.

   ```
   INVALID APPLICATION, TRY AGAIN
   ```

If you get this message, ask the network administrator at your site if you have been validated to use NETOU.

You may optionally want to create a file containing commands to be executed every time you log in. This NOS indirect access file NETOPRP resides in the operator's catalog. Typically, this file defines your command environment. For more information on prologs, see Creating a Prolog in chapter 3 of this manual.

Selecting an MDI or MTI

NOTE

This section is provided for site configurations that have more than one MDI connected to a host. Two MDIs that share only a single NOS host are considered separate catenets.

When you log in to NETOU, your job's connection is switched to NETOU. NETOU responds by connecting your operations station to the default MDI or MTI to receive your network commands and route them through the network. If there is more than one MDI or MTI available for you to select for an operations session, NETOU responds in one of two ways.

- NETOU automatically selects an MDI/MTI for you.
- NETOU prompts you to select an MDI or MTI.

You must also select an MDI if the currently selected MDI breaks its connection with NETOU.

Until an MDI becomes available and you select one, you may only enter the following commands.

DISPLAY_CONNECTED_MDI
DISPLAY_ALARM_HISTORY
ROUTE_ALARM
ROUTE_COMMAND_RESPONSE
SET_COMMAND_MDI
DISPLAY_COMMAND_LIST
DISPLAY_COMMAND_LIST_ENTRY
HELP
DISPLAY_COMMAND_INFORMATION
QUIT
LOGOUT
BYE
GOODBYE
HELLO
LOGIN

All other commands are ignored.

If more than one MDI or MTI is connected to NETOU, you receive a message listing all the MDIs which you can select to connect with the network. The display you receive depends on the number of MDIs and/or MTIs defined at your site. The following is an example of such a message:

```
  STATUS OF CONNECTED MDIs
  NODE    CURRENT     SYSTEM
NUMBER     STATE      TITLE

  043     AVAILABLE   MDI_8A
  044     AVAILABLE   MDI_85
```

If more than one MDI has established a connection with NETOU, as in the previous example, you also receive the following message.

```
More than one MDI available.
Please select an MDI by the following command:
  To select an MDI, type SETCM MDI=MDINAME
  MDINAME is optional, if not specified,
  default is <system title>
```

The command you enter is called SET_COMMAND_MDI (SETCM). The value of <system title> is the MDI to which you are connected if you do not specify an MDI. If you specify no parameter, the first available MDI in the list in the AVAILABLE state, is selected.

A default MDI can be defined in the job statement for NETOU in a host file named NAMSTRT. If the connection with this default MDI is broken, NETOU reselects the default MDI. Unless there is more than one MDI at your site, or if you plan to switch between MDIs, you can use the default MDI. For more information on how to select a default MDI see the NOS Version 2 Installation handbook.

Once you have selected an MDI for communication with the network, you receive that MDI's title in a message sent from that MDI. If you need to check which MDI or MTI you have currently selected, enter the DISPLAY_CONNECTED_MDI (DISCM) command.

NOTE

You receive alarms that are sent through the selected MDI or MTI. If alarms from another catenet are desired, you must select a different MDI using the SETCM commands.

Creating a Prolog

See Creating a Prolog in chapter 3 of this manual if you wish to create a file containing a series of commands that you execute every time you establish a connection.

Prompts

You immediately get the following prompt after logging in to NETOU.

```
NOU/
```

This prompt indicates that you are logged in to NETOU and can begin entering NETOU commands. NOU/ is displayed as a prompt until you select another application, such as IAF.

You immediately get the following prompt after entering the SENCS command.

```
SENCS/
```

This prompt indicates that you are in the SEND_COMMAND_SEQUENCE mode. SENCS mode allows you to send one or more commands to the same system(s) without enclosing the command within a SENC command. The commands you enter following this prompt are sent only to the systems listed in the system parameter of the SEND_COMMAND_SEQUENCE command. SENCS displays as a prompt until you enter ** to exit the SENCS mode.

Paging

Paging allows you to move forward within a display on the terminal screen. You may enable or disable paging using the CDCNET terminal command called CHANGE_TERMINAL_ATTRIBUTES (CHATA). To do this, enter the network command character (NCC) (shown here as a percent sign [%], but the actual NCC may differ for your terminal), and the CHANGE_TERMINAL_ATTRIBUTES command, as in the following example.

```
%CHANGE_TERMINAL_ATTRIBUTES HOLD_PAGE=ON or OFF
```

ON enables paging; OFF disables paging. The default is for paging to be OFF. When paging is ON, to scroll to the next page of text, enter a carriage return or a control character. For more information about the CHANGE_TERMINAL_ATTRIBUTES command and the network command character, see the CDCNET Terminal Interface manual. Instructions for paging at the K display console are provided later in this section.

Displaying Job Status Information

Displaying job status allows you to monitor the progress of your job through the CDCNET network. To do this, enter the network command character (NCC) shown here as a percent sign (%) followed by an e. The actual NCC may differ at your site. The first two lines tell you the current routing of the command responses and alarms. The third line tell you that you are in SENCS mode; that is, if you are in the SENCS mode. A list of the DIs to which the commands are being sent in SENCS mode follows. The last line tells you the current status of your job.

```
%e

Command responses routed to DISPLAY.
Alarms routed to DISPLAY.
You are currently in SEND_COMMAND_SEQUENCE mode
Commands sent to <list of DIs>
You may enter commands.
```

Logout

When you want to log out from the NETOU application (and optionally log in to another application such as IAF), enter one of the following commands. NETOU terminates your current session and prompts you for a new session (if you use LOGIN).

HELLO,application
BYE
LOGOUT
LOGIN,application
GOODBYE
QUIT
QUI

Examples:

The following example logs an operator out of the NETOU application and selects the IAF application.

```
hello,iaf
```

The following example logs an operator out of the current NETOU session and begins a new NETOU session.

```
hello,netou
```

Network Operations from a NOS Host Console

At a NOS host computer console (a CC545 console or a 721 terminal), your interface to CDCNET is through the standard Network Access Method (NAM) host operator interface, the NAM K display. This section focuses on using the NAM K display to access and use NETOU. For background information on host console operations and K displays, see the NOS 2 Analysis handbook.

NETOU K-Display Format

The K-display format used during the NETOU application is identical to the standard NAM K display used for NOS Operations. For further information about K displays, see the NOS Operations handbook. Figure 2-3 shows a typical K display used for CDCNET network operations.

```
K,NAM.
13:30:45 86.01.10
            MID=81   NOS43C/14R8117KD
.......................................................................

    NETWORK_OPERATOR_UTILITY    86/11/10    13.30.45    1478
   WELCOME TO NETWORK OPERATOR UTILITY
    CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1986.
              :

        (data area -- 31 display lines maximum)


READY..     (message line)

ALERTS  (alert line -- a list of applications requesting your attention)

NETOU     SETCM,MDI_80
```

Figure 2-3. NETOU K-Display Format

The NETOU in the lower left corner of the operator entry line indicates that you are logged in to NETOU. To the right of NETOU you see the last command entered. This field contains 40 characters or less. Commands longer than 40 characters are not completely displayed. NETOU uses the K-display alert line and the operator entry line similarly to the standard NAM applications. NETOU does not use the host message line.

The K display has two data areas, left and right. The left data area displays commands, responses, network alarms, and operator prompts. You may display data on this side as a continuous scroll or view it page by page. See the discussion on paging of the K display, later in this chapter. The right data area is not used for NETOU operations in this release.

Login

Use the following procedure to log in to NOS and select NETOU from a host console.

1. To access the NAM K display from the host console, enter the following.

```
K,NAM.
```

2. Select NETOU:

```
K.AP=NETOU
```

3. The NETOU application responds by clearing the left data area and sending the following prompt.

```
READY..
PLEASE ENTER *USERNAME,PASSWORD*,
ENTER VALUES IN ONE LINE, SEPARATED BY COMMAS.
READY..
```

   Enter your user name and password.

```
user name,password
```

   Your user name must be a member of the operating system's default family. For a valid login (a login that is known to the operating system and authorized for CDCNET control access), NETOU responds by sending the following message,

```
USER VALIDATION SUCCESSFUL,UN=<user_name>
```

   and then connects your session to the default MDI or MTI to receive your network commands and route them through the network. If there is more than one MDI or MTI available for connecting to the network, you are prompted (if your site selected the prompting option) to select the MDI or MTI to be used for your operations session. If login is invalid, NETOU reissues the prompt for a valid login. See NOS Version 2 Installation handbook for more information on selecting the prompting option.

4. You receive the status of connected MDIs at your site, and a prompt to choose an MDI, if more than one MDI is available.

```
  STATUS OF CONNECTED MDIs
  NODE    CURRENT     SYSTEM
NUMBER     STATE      TITLE

  043     AVAILABLE   MDI_8A
  044     AVAILABLE   MDI_85
```

   If more than one MDI has established a connection with NETOU, as in the previous example, you also receive the following message.

```
More than one MDI available.
Please select an MDI by the following command:
  To select an MDI, type SETCM MDI=MDINAME
  MDINAME is optional, if not specified,
  default is <system title>
```

5. Enter SET_COMMAND_MDI (SETCM). The value of <system title> is the default MDI to which you are connected if you do not specifically select an MDI. The default MDI is the first MDI listed as AVAILABLE. If you enter only SETCM with no parameter, then the first DI in the list is selected.

   You receive a message showing the current user name in effect when the K display is reassigned after you have logged in.

   ```
   YOU ARE CURRENTLY LOGGED IN AS UN=user_name
   ```

   You may also see alarms that have been sent since a DISPLAY_ALARM_HISTORY command was issued, and a notification of the current operator state, such as command in progress.

You may wish to create a prolog, a file containing a series of commands that you execute every time you establish a connection. For information on how to create a prolog, see Creating a Prolog, in chapter 3 of this manual.

Logout

To log out from NETOU, enter any of the following logout commands:

K.LOGIN
K.LOGOUT
K.GOODBYE
K.BYE
K.QUIT
K.QUI
K.HELLO

All the above logout commands perform two actions: terminate the current session and begin a new session.

After logout, a login prompt is displayed. You must type K.* to return the K display to NAM control. Once you log out, alarms issued by the network are discarded. Any commands you sent prior to the logout may or may not complete, but you do not receive responses to these commands.

Exiting and Resuming NETOU Sessions

At the host console, you may exit NETOU without logging out of NETOU completely. To do this, enter:

K.*

K.* returns the K display to NAM control. NETOU remains active and retains your login. NETOU continues to monitor the network for alarms, even though you are not currently using the application. If new alarms occur during this time, the following message appears on the K display alert line.

```
NETOU
```

Any alarms received at your operations site can be displayed after you resume using NETOU.

To resume your operations session, enter:

```
K.AP=NETOU
```

NETOU returns the following message.

```
YOU ARE CURRENTLY LOGGED IN AS UN = <username>
```

Immediately following the above message, the most recent alarms are automatically displayed. Most recent alarms are those that have been sent since you last entered the DISPLAY_ALARM_HISTORY command. After alarms are displayed, you receive status information (see Displaying Job Status Information, in this chapter), followed by either the READY.. prompt or information that a command is in process. The NETOU prompt is cleared from the information alert line.

Prompts

Common prompts at the K display include the following.

Prompt	Description
READY..	Command entry is allowed.
MORE DATA..	Page wait is on and more pages (screens) of data exist. Enter K.+ to see the next page of data or K.- to turn page wait off and see the rest of the data.
REPEAT..	You entered a command before the NETOU application was ready to receive it. Wait until you see the READY.. prompt, and reenter the command.
COMMAND TOO LONG	The command you are entering is too long for the K display (see Continuing Commands later in this chapter).
LINE TOO LONG	The line of data you are entering is too long for the K display (see Continuing Commands later in this chapter).

Most prompts are displayed at the bottom left corner of the K display's left data area. The REPEAT.. prompt is displayed at the right margin of the operator entry line.

Paging

Some command responses fill more than one screen of the K display. When page wait is on, the MORE DATA.. prompt indicates this. You may view additional screens of data (also known as paging) and control paging of the data areas by entering the following commands. You may enter commands to turn paging on and off for the left data area. By default, paging is off at the K display.

Command	Description
K.+	Turns paging on for left data area. When you first enter NETOU mode, paging for the left data area is off. Once paging is on, NETOU only displays one page at a time of a multipage response. Multipage responses are indicated by the MORE DATA.. prompt. You may scroll to the next page by again entering K.+.
K.–	Turns paging off for the left data area. If you enter K.– instead of K.+ when the MORE DATA.. prompt appears, paging is shut off. The screen immediately displays all responses, and MORE DATA.. is not displayed again.

If you change page wait from off to on or on to off, the success response is as follows:

```
PAGE ACCEPTED.
```

K-Display Console Entry Restrictions

All commands at the K display are entered as follows:

K.command

The K. prefix is required. The syntax used for the command portion is the same as that used at an interactive terminal. See Common Network Operations Features in this chapter for more information on command syntax.

Normally, once the K display is active, the K. is automatically generated each time you enter a command. If you cancel the automatic feature by pressing the erase (left blank) key on the system console, you can restart the automatic process again by reentering the K. before the next command. Enter a carriage return to indicate the end of a command.

Entering Characters Not Supported at a NOS Host Console

NETOU commands use a subset of the syntax for NOS/VE SCL commands. SCL uses the ASCII character set, which has characters the NOS host console (CC545 and 721) does not support. On the NOS host console, you must type two characters, or an escape sequence, to designate the ASCII characters not supported on the console.

On the NOS host console screen, unsupported ASCII characters are designated by other characters. For a character which represents more than one ASCII character when displayed, such as the asterisk (*), the only way to know which ASCII character it represents is by the display's context. Table 2-1 shows escape sequences for unsupported ASCII characters and how these characters are represented on the console screen.

The following example compares command entries made at a terminal that supports the full ASCII character set with the same entries made at a NOS host console using the escape sequences. In this example, the hyphen is used rather than the /0 sequence to represent the underscore character.

ASCII terminal entry:

```
send_command command='display_hardware_status',system=north_tdi_1
```

System display console entry:

```
SEND-COMMAND COMMAND=/*DISPLAY-HARDWARE-STATUS/*,SYSTEM=NORTH-TDI-1
```

Table 2-1. NOS Host Console Escape Sequences and Displays

Character	Name	Escape Sequence On Keyboard	Displayed On Screen As:
^	Circumflex	/1	/1
"	Quotation Marks	/2	/2
#	Number Sign	/3	/3
$	Dollar Sign	/4	/4
@	Commercial At	/5	/5
;	Semicolon	/6	/6
?	Question Mark	/7	/7
{	Opening Brace	/8	/8
}	Closing Brace	/9	/9
_	Underline	Hyphen (-) or /0	-
[	Opening Bracket	/(	/(
]	Closing Bracket	/)	/)
>	Greater Than	/+	/+
<	Less Than	/=	/=
'	Apostrophe	/*	/*
/	Slant	//	/
!	Exclamation Point	None	.
%	Percent Sign	None	*
&	Ampersand	None	+
\	Reverse Slant	None	*
`	Grave Accent	None	*
|	Vertical Line	None	*
~	Tilde	None	*
:	Colon	/,	.
-	Minus, Hyphen	/-	-
a..z	Lowercase	/A../Z	A..Z

Continuing Commands

The K display does not accept input of more than 50 characters after the K. If you enter a command that goes over this limit, you receive one of the following prompts in the lower left corner of the console screen.

```
LINE TOO LONG

COMMAND TOO LONG
```

When this happens, the command entry is not processed. You may not enter anything else until you clear the entry by one of the following methods.

- Press the backspace key repeatedly, until you have fewer than 50 characters.
- Erase the entry by using the left blank (erase) key on the system console keyboard. Then reenter the command starting with the K.

To enter command strings that are longer than 50 characters, use the continuation symbol, the ellipsis (..), before you enter the 48th character, and enter a carriage return. Continue the command on the next line. The following examples show how to enter a multiple-line command from a system console. Assume that the hyphen represents the underscore character and that each line ends with a carriage return.

```
K.SEND-COMMAND ..
K.C=/*DISPLAY-LINE-STATUS LINE-NAME=(COMPSCI-02 ..
K.ENGINEERING-PORT-1 ENGINEERING-PORT-2 ..
K.ENGINEERING-PORT-3)/* SYSTEM=NORTH-TDI-2
```

Command Syntax for NOS NETOU

This section describes the special syntax rules and the process used for sending CDCNET network operations commands from your operations station to the network in a NOS-based operations environment.

In a NOS environment, NETOU has the following types of commands.

- Commands executed on the host (session control commands).
- Commands executed in the MDI through which you are communicating with the network (session control commands).
- Commands executed in DIs throughout the network (network commands).

All these commands follow a subset of the NOS/VE SCL syntax (see Command Syntax later in this chapter). All network operations commands share the following properties in a NOS environment.

- Lowercase letters are interpreted as uppercase letters, with the exception of lowercase strings enclosed within single quotation marks (').
- Entering more than one network operations command per entry line is prohibited.

Some commands require parameters, such as FILE_NAME, that are passed on to NOS. The values allowed for these parameters have the same syntax and limits as those used in the NOS command language.

Entering NETOU Commands on NOS

NETOU commands are valid only within a NETOU session. The session begins when you select NETOU. The session ends when you log out of NETOU.

Network commands are embedded within SEND_COMMAND, which is interpreted by the Operator Support Application (OSA) in the MDI through which you are communicating with the network.

SEND_COMMAND (NOS Version)

To send network commands through the network, use SEND_COMMAND (SENC), transmitting the network commands to the DI you specify. Except for the session control commands described later in this manual, you must embed all network commands in a SEND_COMMAND. To use this command, enter:

```
SEND_COMMAND COMMAND=string,SYSTEM=name
```

COMMAND (C) is the network command to be sent to the DI specified with the SYSTEM parameter. Enter the command as a string value enclosed by apostrophes (').

NOTE

If the network command you are sending contains any apostrophes, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized. Otherwise, NETOU assumes the embedded apostrophe signals the end of the network command, and errors result.

SYSTEM is the logical or physical DI name or list of DI names to which you want to send the command. SYSTEM is an optional parameter. If you omit this parameter, the last DI to which you sent a command is used. If SYSTEM is omitted on the first SEND_COMMAND you use after you log in to NETOU, the selected MDI is used as the value for the SYSTEM parameter. The SYSTEM parameter may specify a maximum of 15 systems to which you want to send a network command with a single SENC command. If a network command is sent to more than one DI, a response from each DI must be received for the command to complete. The SYSTEM parameter is optional for SEND_COMMAND in NOS environments, but required for SEND_COMMAND in NOS/VE environments.

NOS SEND_COMMAND Examples

1. Enter the following command sequence to stop traffic on a communications line connected to a DI named TDI_3. The actual command to stop communications traffic is enclosed within a SEND_COMMAND that specifies the DI (TDI_3) to which the line is connected.

   ```
   send_command command='stop_line line_name=line3',system=tdi_3
   ```

2. Use the following SEND_COMMAND command to send a DISPLAY_LINE_STATUS command sent to the same DI as in example 1 (TDI_3). In this example, the SYSTEM parameter can be omitted, since the previous SEND_COMMAND specified TDI_3.

   ```
   send_command command='display_line_status'
   ```

NETOU Session Control Procedures 3

NETOU Session Control Procedures 3

This chapter contains descriptions of NETOU session control procedures. Session control is a term used to describe the set of actions you take to define, change, and control the online environment for your CDCNET network operations sessions. Examples of session control include routing command responses and alarms to files, and executing files of CDCNET commands.

Session control commands differ from other network operations commands because they are not sent to DIs. They define your operations setup and are not enclosed within SEND_COMMAND.

This chapter is divided into two sections: Session Control on NOS/VE (for NOS/VE-based operations) and Session Control on NOS (for NOS-based operations). Each section provides instructions for doing session control activities.

NOTE

For a complete description of the commands in this chapter, see the CDCNET Commands Reference manual.

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual. The CDCNET Network Management Station has a utility similar to NETOU.

Session Control on NOS/VE

This section describes how to use commands and functions to control your CDCNET network operations sessions in a NOS/VE environment. In a NOS/VE environment, most of the CDCNET operations session control is done through standard SCL functions, commands, and services on NOS/VE. If standard SCL functions, commands, or services are used to perform any activities, they are referred to in the text, but not described in detail. You are directed to the appropriate NOS/VE SCL manual for more information.

SCL Functions for NETOU Sessions

NETOU provides the following SCL functions to help you perform iterative operations and to use NETOU commands in combination.

$NORMAL_RESPONSE

This function returns a value of TRUE if a normal response was received from the last CDCNET command sent by the SEND_COMMAND command.

The command format is:

```
$NORMAL_RESPONSE(name)
```

where *name* is the name of the system for which the response is to be checked. This parameter is always optional. If the last CDCNET command was sent to more than one system and the *name* parameter is omitted, then a value of TRUE is returned only if all of the responses were normal.

$RESPONSE_IDENTIFIER

This function returns the command response identifier from the response to the last CDCNET command sent by the SEND_COMMAND command. Response identifiers are integers in the range 33000..65535. The meaning of a specific value is described in the online CDCNET Diagnostics Messages manual.

The format for this function is:

```
$RESPONSE_IDENTIFIER(name)
```

where *name* is the name of the system for which the response is to be checked. This parameter is optional if a command is sent to only one CDCNET system. If the last CDCNET command was sent to more than one system, then the *name* parameter is required.

$MATCHING_NAMES

This function returns a list of CDCNET system names matching a name pattern. The list of names is assigned to an SCL array variable that is then used as the value for the SEND_COMMAND parameter that sets the destination for a series of CDCNET commands. The name pattern may contain wildcard characters. For this release of CDCNET, wildcards are supported for the $MATCHING_NAMES function only. (See Wildcard Characters, next, for more information.)

The format of the function is:

```
$MATCHING_NAMES(string)
```

where **string** is a string representing the pattern to be matched. This is a required parameter. Enclose the string value within apostrophe characters.

Example:

```
$MATCHING_NAMES('$*')
```

Wildcard Characters

Optional wildcard characters allow you to address a command to CDCNET systems using names that match a specific name, as modified by a wildcard character. Names used as the destinations for network commands may be modified by the following wildcard characters.

Character	Description
?	Represents any single character.
*	Represents any string of characters.
[ ]	Represents any one of a set or range of characters collated in the ASCII character set. For example, [3ab4] represents any one of the character set 3, a, b, or 4. The abbreviation [3-6] represents any one of the characters 3, 4, 5, or 6.

SCL Procedures for NETOU Sessions

You can create and use SCL procedures that use the functions described in this section to enhance your NOS/VE NETOU environment. For example, you could create a procedure that uses the $MATCHING_NAMES function to send a command to a set of DIs that match a name modified by wildcard characters.

Session Control Activities

This section contains instructions for using NOS/VE-based session control commands and functions to set up and control your operations sessions.

Creating a Prolog

A prolog is a file containing a list of commands that are executed each time an activity is initiated. You can create a prolog specifically for your NETOU sessions that are executed every time you access NETOU. A prolog is not required for a successful invocation of NETOU. The commands you put in the prolog are up to you. Any NETOU or NOS/VE commands may appear in the file. For example, the ACTIVATE_ALARMS command must be entered any time you invoke NETOU if you want to enable alarm reporting at your operations station. Instead of entering this command every time, you could put it in your prolog to automatically enable alarm reporting whenever you invoke NETOU.

The default prolog file name is $USER.NETWORK_OPERATOR_PROLOG. However, you may define alternate prologs and put them in any catalog you can access through a normal NOS/VE file reference. When you invoke NETOU with the NETOU command, use the PROLOG parameter to specify the file reference for your prolog.

During NETOU sessions, other files called command files can be used to simplify command entry. The next section provides information on command files.

Building Command Files

Command files contain CDCNET network operations commands (both session and network control commands) as well as any other NOS/VE commands. You can use the NOS/VE command INCLUDE_FILE to process a command file. The INCLUDE_FILE command causes the text of a file to logically replace the occurrence of the INCLUDE_FILE command. The commands in the specified file are then processed. Each line of the command file is executed as if it were an individual command you typed in at your operations site. For more information on the INCLUDE_FILE command, see the NOS/VE Commands and Functions manual. You may build command files to perform session and network control activities. A break sequence terminates command file processing.

Command files can be an efficient way to send commands and save keyboard entry, since you can group several commands that perform a single activity together in a file. Once a command file is created and saved, when you need to perform an activity such as redefining a line, you specify the file with the INCLUDE_FILE command rather than entering all the commands individually. Chapter 4 describes network operations activities and the commands that perform the activities. You can build command files to perform the activities described there.

You can also use command files to send a command to several DIs. The command file would have the same command on every line, but the DI name specified on the SEND_COMMAND would differ for each line.

Writing and Executing Command Files

The following procedure makes use of the concepts for managing NOS/VE files. For more information, see the NOS/VE System Usage manual. This procedure also assumes you can use the full screen editor for NOS/VE.

1. After logging in to NOS/VE, enter NETOU by typing in the following:

```
/netou
```

2. Create and edit a file using the full screen editor by entering the EDIT_FILE command. When creating the file, you must specify the FILE_CONTENT and FILE_PROCESSOR parameters. The FILE_CONTENT = LEGIBLE parameter permits the file to contain character data. The FILE_PROCESSOR = SCL specifies that SCL processes the data.

```
edit_file file=di_status
```

3. You may put any NOS/VE session control commands and CDCNET network control commands in the file. For network control commands, be sure to enclose the commands within the SEND_COMMAND command. You can also enter other NOS/VE commands in the file.

```
"File DI_STATUS contains the DISPLAY_DI_SYSTEM_STATUS command."
"When the file is executed, the status command will be sent to"
"the three DIs specified in the file by the SEND_COMMAND."

senc c='disdss',s=mdi1
senc c='disdss',s=tdi1
senc c='disdss',s=tdi2
```

NOTE

Always note the command file's purpose, either in the file itself (as a comment) or in your records. This is important if you have many command files or several versions of a command file.

4. To execute the command file, use the INCLUDE_FILE command and the file name parameter. For descriptions of the other parameters, see the INCLUDE_FILE description in the NOS/VE System Usage manual.

```
include_file file=di_status
```

The commands in the DI_STATUS file are sent to the appropriate DIs, where they are executed.

In this example, the command file DEFINE_ETHERNET is a standard set of commands used to redefine an Ethernet network solution. Parameter values are left blank so the file can be copied and parameter values can be specified.

```
"File DEFINE_ETHERNET"
"This file is a template file of network operations commands"
"that can be copied and used to define an Ethernet network solution."
"Insert the appropriate parameter values where indicated."
"Not all optional parameters are shown.  If other parameters are added"
"to the command being sent, they must be placed within the final"
"apostrophe character."

Send_command Command='Stop_network Network_name=      'System=

Send_command Command='Cancel_ether_net Network_name=  ',System=

Send_command Command='Define_ether_trunk Slot=  ,Trunk_name=',System=

Send_command Command='Define_ether_net,..
  Trunk_name=      ,Network_ID=       Network_name=       ',System=

Send_command Command='Start_Network Network_name=      ',System=
```

A command file is useful in this situation because defining and starting a network solution involves defining and starting the network at two places. Once you have created a file of commands to define and start a network solution, you can duplicate and use the file to define and start the network solution on each DI affected by the definition change. This command file includes comments that describe the file's use.

Using SCL Procedures

An SCL procedure is a series of SCL statements that perform a specific task. Because SCL allows parameter substitution, SCL procedures are easier to use to perform routine network management activities. You can develop your own SCL procedures for your particular site.

Activating and Deactivating Alarms

Every DI generates alarms ranging from informative messages to indications of software failures. By default, these alarms are not sent to your operations station unless you explicitly activate them. To activate alarms so that they are displayed at your operations station, you must activate transmittal from the host to your station any time you invoke NETOU, by entering the ACTIVATE_ALARMS command. To ensure that this command is entered every time you invoke NETOU, include ACTIVATE_ALARMS in your user prolog (see Creating a Prolog in this section). Then, when you enter the NETOU command to invoke NETOU, alarms are activated.

Deactivate alarms by shutting off the transmittal of alarms from the host to your operations station. To do this, enter the DEACTIVATE_ALARMS command.

For NOS/VE CDCNET operator environments, all alarms received at the operations station are displayed when alarms are activated. Either all DIs in the network send alarms to you, or no DIs send alarms. There is no way to selectively deactivate an individual DI's alarms using session control commands. Instead, you must send the network control command CANCEL_SOURCE_ALARM_MESSAGE to the DI and specify the appropriate alarm message numbers. This command turns alarm messages off for all operators, because it directs the DI not to send the alarm.

Routing Command Responses and Alarms

You can route command responses and alarms to files other than your display screen using standard NOS/VE files and commands. Routing responses and alarms to files can help you keep a record of responses and alarms. You can review the files and print them, if necessary. Routing is helpful with lengthy responses, such as the responses to the display status and configuration network control commands, which may return several screens of data.

To route responses and alarms to files, use the SCL command CREATE_FILE_CONNECTION. See the NOS/VE Commands and Functions manual for a complete description of this command. CREATE_FILE_CONNECTION establishes a connection between one of the standard NOS/VE files and one or more files. Any data written to the standard file is also written to the file you specify. The allowed standard file names include the following.

$ECHO
$ERRORS
$INPUT
$LIST
$OUTPUT
$RESPONSE

Routing Responses

Normal command responses are written to the file specified on SEND_COMMAND. The default output file is the standard NOS/VE file $OUTPUT. Error responses are written to standard NOS/VE file $RESPONSE. Use the CREATE_FILE_CONNECTION command to connect a file to these standard files. If you only want a file of error messages, specify $RESPONSE.

NETOU commands and any NOS/VE commands you enter are written to standard file $ECHO. For a complete record of your operations sessions which include both commands and responses, use the CREATE_FILE_CONNECTION command to connect a file to $OUTPUT, $RESPONSE and $ECHO. You can use the standard job log file ($JOB_LOG) to serve as the file to which all commands and responses are written. The job log adds a date and time stamp to the commands and responses. By default, $RESPONSE is connected to $JOB_LOG.

Routing Alarms

All alarms are written to the file specified on ACTIVATE_ALARMS. The default output file is the standard file $OUTPUT. For an alarm history file, use the CREATE_FILE_CONNECTION command to connect another file to $OUTPUT, or to any other file you specify as the one to receive alarm output on. You can write the alarms to the same file to which responses are written.

Accessing Response and Alarm Files

Use the standard NOS/VE commands for accessing and displaying the files to which responses and alarms are written. If you write responses and alarms to $JOB_LOG, use the DISPLAY_LOG command to display the job log.

Responding to Alarms

Check the online CDCNET Diagnostic Messages manual for the description of the alarm you have received and the suggested actions for each message.

Alarms may also be messages to you from a terminal user. Respond to a message from a terminal user by the same line name listed in the alarm, using the WRITE_TERMINAL_MESSAGE command.

Session Control on NOS

This section describes how to use commands to control your CDCNET network operations sessions in a NOS environment.

You must be validated on the family in which the NETOU application executes. If you are not validated on that family, you cannot read the prolog and command files, and you cannot write routing files.

Creating a Prolog

A prolog is a file containing a list of commands that are executed each time an activity is initiated. You can create a prolog for your NETOU sessions that is executed each time you access NETOU. A path to CDCNET must be available at the time you access NETOU. You can put any NETOU command in your prolog file. Typically, your prolog contains the CDCNET commands to establish your command environment. Rather than entering these commands every time you access NETOU, put the commands in your prolog to establish your command environment whenever you invoke NETOU. You could also include a SEND_COMMAND_SEQUENCE (SENC) command in your prolog. This eliminates typing because you need not enclose each command within the SENC command.

The default prolog file, NETOPRP, is a NOS indirect access file which resides on your operator's catalog. However, you can specify a different prolog file name on the APPSW command when you log in to NETOU. The file name on the APPSW command is then used as your prolog file.

The APPSW command has the following format.

APPSW,AP=NETOU,Z.PROLOG=<name>,MDI=<name>

AP

The application to which you want to connect (in this case, NETOU).

Z

Indicates additional data follows the period. The data is saved and passed to the application you specified with the AP parameter.

PROLOG (P)

Specifies the name of the permanent file to be used as your user prolog file. Follow the NOS file naming conventions when naming your prolog file. If this optional parameter is omitted, the default file, NETOPRP is used as the prolog file.

MDI (M)

Specifies the system title of the MDI to which the operator's session is to be switched. Using this parameter is equivalent to logging into NETOU and entering the SET_COMMAND_MDI (SETCM) command. However, using this optional parameter causes suppression of the copyright banner and MDI selection message you would otherwise receive after logging in to NETOU. If you specify an MDI on the APPSW command and that MDI subsequently fails, NETOU logs you out and returns the connection to IAF.

The PROLOG and MDI parameters are positional parameters. If you do not use the keywords, specify the prolog file name followed by the MDI name. If you do not use the keywords and if you do not specify the prolog file name, substitute a comma for the prolog file name, followed by the MDI name.

During NETOU sessions, you can use other files, called command files, to simplify command entry. The next section provides information on command files.

Building Command Files

Command files are files containing CDCNET network operations commands (both Session Control commands and the commands that monitor, control, and configure DIs). You can build command files to perform session and network control activities. Each command in the file is executed as if it were an individual command you typed in at your operations site. A break sequence terminates command file processing.

Command files can be an efficient way to send commands and save keyboard entry, since several commands that perform a single activity can be grouped together in a file. Once a command file is created and saved, when you need to perform an activity for which the command file was created, you can call the command file and execute it using the EXECUTE_COMMAND_FILE command, rather than entering all the commands individually. Network operations activities and commands that perform the activities are described in chapter 4. You can build command files to perform the activities described there.

NOTE

Some commands and procedures used to perform network operations activities are not a part of NETOU, but run under another application. You may not include commands and procedures that are not CDCNET network operations commands in command files. Commands and procedures that are described in this manual but are not allowed in CDCNET network operations command files include:

- Network_Logfile_Termination Utility (NLTERM).
- Network_Logfile_List (NLLIST).
- All Network Performance Analyzer (NPA) commands and procedures used to obtain network statistics.

Writing and Executing Command Files

The following procedure assumes that you have access to an editing program, such as NOS Full Screen Editor (FSE). Command files can be created at either a host console or an interactive terminal. However, because interactive terminals with full screen interface are better suited to file editing, this procedure is geared toward an interactive terminal using FSE.

1. CDCNET command files must be written in the NOS 6/12 ASCII character set. To ensure this, enter the NOS ASCII command prior to accessing FSE.

   ```
   ascii
   ```

2. Create a NOS local file by entering FSE and the name of the new file.

   ```
   fse,newfile
   ```

3. Using FSE, enter the appropriate session and network commands in the file.

   NOTE

   The commands EXECUTE_COMMAND_FILE, INCLUDE_FILE, and SET_COMMAND_MDI cannot be used in command files. To put comments in the command file, enclose the comment text in quotation marks.

4. Make the command file an indirect access permanent file using the SAVE command.

   ```
   save,newfile
   ```

   NOTE

   You should make a note of the command file's purpose, either in the file itself or in your records. This is important if you have many command files or several modified versions of a command file.

5. Exit the NOS Interactive Facility (IAF) and enter NETOU by entering the following:

   ```
   /bye,netou
   ```

6. Test the file by executing it using the EXECUTE_COMMAND_FILE command.

   ```
   execute_command_file file=newfile,user_name=name
   ```

Provide the name of the command file you want to execute. USER_NAME is optional. Use it if the command file is not in your permanent file catalog, but under another user name. In that case, the file must be public or semiprivate, as you must have permission to access the file.

The following command file sends a set of display status commands to a list of three DIs, MDI1, TDI1, and TDI2.

```
"File STATUS displays status of DI hardware and software."
sencs s=(mdi1, tdi1, tdi2)
display_di_system_status,
display_hardware_status,
display_line_status,
display_network_status,
display_software_load_status,
display_directory_status,
**
```

The following command file, DEFETH, is a standard set of commands used to logically reconfigure an Ethernet network solution. Parameter values are left blank so the file can be copied and parameter values can be specified. A command file is useful in this situation because defining and starting a network solution involves defining and starting the network at two places. Once you create a file of commands to define and start a network solution, you can duplicate and use it to define and start the network solution on each DI affected by the definition change. This command file includes comments that describe the file's use.

```
"File DEFETH"
"This file is a template file of network operations commands"
"that can be copied and used to define an Ethernet network solution."
"Insert the appropriate parameter values where indicated."
"Not all optional parameters are shown.  If other parameters are added"
"to the command being sent, they must be placed within the final"
"apostrophe character."

Send_command Command='Stop_network Network_name=      'System=

Send_command Command='Cancel_ether_net Network_name=  ',System=

Send_command Command='Define_ether_trunk Slot=  ,Trunk_name=',System=

Send_command Command='Define_ether_net,..
  Trunk_name=      ,Network_ID=       Network_name=       ',System=

Send_command Command='Start_Network Network_name=      ',System=
```

The following EXECUTE_COMMAND_FILE example executes a file called TRMSTAT, that starts collection and reporting of line statistics. The file TRMSTAT is under another user name, so an alternate user name is specified with the command.

```
EXECUTE_COMMAND_FILE FILE=TRMSTAT,UN=ZELDA
```

Routing Command Responses and Alarms

You can route command responses and alarms to a file using the ROUTE_COMMAND_RESPONSE and ROUTE_ALARM commands. Routing of responses and alarms allows you to review responses, retain them in a NOS permanent file, and print the file to more thoroughly review the responses. Routing is helpful with lengthy responses, such as status and configuration displays, which may return several pages of data.

To route responses, enter:

```
ROUTE_COMMAND_RESPONSE FILE = (file_name,DISPLAY) or DISPLAY or file_name
```

To route alarms, enter:

```
ROUTE_ALARM FILE = (file_name,DISPLAY) or DISPLAY or file_name
```

Specify a file name as the file to receive the responses or alarms. This file must be a NOS direct access permanent file. If the file does not exist when the command is executed, a new file is defined. If the file does exist, responses or alarms are appended to the end of the file. If you enter DISPLAY, command responses or alarms are routed to your operations station. If you enter DISPLAY without any parameters, command responses or alarms are routed to your operations terminal (DISPLAY is assumed). If you specify a file name but do not enter DISPLAY, command responses or alarms are not routed to your operations station. At the start of your session, routing of responses to your operations station (DISPLAY) is assumed.

You may simultaneously route command responses or alarms to your display and to a file by specifying both DISPLAY and another file name as a list with the command. You may also route command responses and CDCNET alarms to the same file.

When requesting the status of several DIs and lines, you could create a file called NSTATUS to receive the status responses, and route the responses to NSTATUS by entering:

```
route_command_response file=nstatus
```

The following command example directs all alarms to a file named OPALARM and to the operations station.

```
route_alarm file=(opalarm,display)
```

Accessing Routed Responses and Alarms

To access files containing CDCNET command responses and alarms, log into IAF or switch to your IAF connection by the CHANGE_WORKING_CONNECTION terminal user command, if you have established multiple connections at your operations station. Use the NOS command ATTACH to attach the file, and the Full Screen Editor to view the file. You may also route the file to a printer using the NOS command ROUTE. See the NOS Reference Set, Volume 3 for the format of the ROUTE command.

Displaying Alarm Environment

The DISPLAY_ALARM_ENVIRONMENT command shows the current alarm reporting setup for your operations station.

```
display_alarm_environment

Alarm Environment
  Community               Alarm Status
CATENET                   Enabled

 Disabled Systems
-None-
```

Changing Alarm Environment

To change the alarm reporting setup for an operations station, enter the CHANGE_ALARM_ENVIRONMENT (CHAAE) command. This command changes the list of DIs that send alarms to you. The CHANGE_ALARM_ENVIRONMENT command also enables alarms.

To shut off alarms from a DI, enter:

```
CHANGE_ALARM_ENVIRONMENT DISABLE_SYSTEM= DI name or names
```

To turn alarms from a Dl back on, enter:

```
CHANGE_ALARM_ENVIRONMENT ENABLE_SYSTEM=DI name or  names
```

NOTE

The CHANGE_ALARM_ENVIRONMENT command is effective only for the operator who enters the command. If there is more than one network operations station active at your site, the alarms still go to the other operators. If you want to turn off alarms for all operators, cancel the source alarm messages at the individual DIs using the CANCEL_SOURCE_ALARM_MESSAGE command.

There are two other commands that can be used to activate and deactivate receipt of all alarms from *all* DIs at an operations station: ACTIVATE_ALARMS and DEACTIVATE_ALARMS. You cannot selectively enable or disable alarms with these two commands; use CHANGE_ALARM_ENVIRONMENT and specify the DIs for which you want to activate alarms.

The ACTIVATE_ALARMS command activates receipt of alarms from *all* DIs in the catenet at an operations station. The effect of ACTIVATE_ALARMS is the same as using the CHANGE_ALARM_ENVIRONMENT command to enable all alarms in the CATENET community of DIs. On NOS, alarms are activated by default. You do not need to use an ACTIVATE_ALARMS command to enable alarm reporting at your operations station at the beginning of your NETOU session.

The DEACTIVATE_ALARMS command deactivates receipt of alarms from *all* DIs in the catenet. The effect of DEACTIVATE_ALARMS is equivalent to using CHANGE_ALARM_ENVIRONMENT to disable all alarms in the CATENET community of DIs.

Restoring Alarm Environment

Use the CHANGE_ALARM_ENVIRONMENT command to add DIs back to the list of DIs that report alarms to you, or use the RESTORE_ALARM_ENVIRONMENT command. The RESAE command restores all DIs to the list of DIs reporting alarms to you. This list of DIs was originally defined at the beginning of your operations session.

Displaying Alarm History

The DISPLAY_ALARM_HISTORY command displays the alarms received at your operations station since the start of your NETOU session.

```
DISPLAY_ALARM_HISTORY DISPLAY_OPTION=option
```

The options for this command are LAST, PAGE, and ALL. LAST displays all alarms received since the last DISAH command was entered. PAGE displays the last page of alarms received. ALL displays all alarms received in the alarm history buffer, which is limited by buffer size to 50 lines of display. If the buffer receives more than 50 lines of display, new lines of display are written over the oldest alarms in the file. Because there is a blank line between each alarm, you may see only 34 nonblank lines of text.

For example,

```
display_alarm_history
```

returns this display.

```
ALARM HISTORY REPORT

****** ALARM FROM MTI_83            85/10/10  13.38.51     619
--ERROR--   Line: LINE31 down, connection timer expired

****** ALARM FROM MTI_83            85/10/10  13.38.55     202
--ERROR--   Line: LINE23 down, auto-recognition failed

****** ALARM FROM MTI_83            85/10/10  13.40.28    202
--ERROR--   Line: LINE23 down, auto-recognition failed
```

Responding to Alarms

Check the online CDCNET Diagnostic Messages manual for the description of the alarm you have received online and the suggested actions for each message. Alarms may also be messages to you from a terminal user. If the alarm is a message from a terminal user, send a message back to the terminal user by the same line name listed in the alarm using the WRITE_TERMINAL_MESSAGE command.

NETOU Network Control Procedures 4

NETOU Network Control Procedures 4

This chapter contains information on how to monitor and control a network. This includes the following:

- Recordkeeping
- Network Operation Commands
- Advanced Activities

Recordkeeping

Keeping track of the network's components, their locations, and their maintenance schedule is an important part of network operations.

Your recordkeeping system should include:

- A diagram of the network's physical layout. The diagram should note the location of all equipment at your site (mainframes, DIs, Ethernet cables, communication lines, hardwired [dedicated] terminals, dial-up [switched] lines, and other network equipment).
- A current list of the logical names assigned to network components. The names for physical components (DIs, network solutions, communication lines) should be shown on the network diagram. When configuration changes or replacements are made, be sure to update this list. You can use the following commands to generate lists of the current logical names and titles defined for the network: DISPLAY_LOGICAL_NAMES, the $MATCHING_NAMES function on NOS/VE, and DISPLAY_CATENET_TITLES on NOS.
- The channel number and mainframe ID for the mainframe connected to an MDI or MTI.
- A list of the serial numbers assigned to DIs. These should also be included on the network diagram.
- Dial-up connections and their baud rates.
- A list of all ports for each DI and each line connected to the DI.
- Maintenance records for all DIs including diagnostic results, repairs, and replacements; problems reported to operator; and records of maintenance personel visits.
- Network Performance Analyzer (NPA) reports.
- A diagram of Service Access Control restrictions (see the CDCNET Configuration Guide for more detailed information).

NOTE

If network management Service Access Control is not permitted on a network solution, you may not be able to send commands to certain DIs. Ask the network analyst which network management is permitted.

- The DI, for DIs supported by NOS hosts, that contains the catenet's master clock (from which all other DIs set their clocks). The location of the master clock is determined during configuration when the functions for each DI are defined in the DI's configuration file. A DI that contains the master clock is known as a clocking system. For DIs supported by NOS/VE systems, the master clock is configured in a NOS/VE host.

NOTE

If you do not have a record of which DI contains the master clock, send a DISPLAY_SYSTEM_OPTIONS (DISSO) command to each DI in the network. Specify with the DISPLAY_OPTION (DO) parameter that you want the DI to return a display of whether or not it is a clocking system.

```
SEND_COMMAND COMMAND='DISSO DO=CLOCKING_SYSTEM',SYSTEM=di_name
```

The DI that contains the master clock for the network returns

```
Clocking system=yes.
```

Mark the location of the master clock in your records and on the network diagram.

You may find it helpful to attach tags to your DI's listing:

- Mainframe (where applicable, as in MDIs and MTIs).
- Mainframe channel number (where applicable).
- Ethernet trunk (where applicable, as in MDIs and TDIs).
- DI type (MDI, MTI, TDI, NDI, RTI) or gateway DI.
- DI serial number.
- DI system ID.

Such information is helpful for people at your site who are unfamiliar with CDCNET hardware, and for you when dealing with CDCNET network problems over the phone.

You can develop an online recordkeeping database of information about network components such as DIs, circuits, lines, ports, locations, and logical names, by using the configuration files for the DIs. Include the previously listed information as comments in the configuration files for your DIs. You can also include comments such as the system ID, the DI's location, and the original date of installation and of subsequent configuration changes. Print copies of the configuration files regularly and arrange them in a binder.

It is important to update the map and the database regularly, particularly when configuration changes, problems, and repairs occur. If you are one of several network operators at your site, be sure to keep each other informed about changes to the records.

Network Operation Commands

To perform basic network activities, you must know how to execute the various types of network operation commands. The command types are shown in the following list and are further explained in this chapter.

NOTE

For a complete description of the commands used in the following procedures, see the CDCNET Commands Reference manual.

- Display status commands
- Communications control commands
- Operator messages commands
- Clock management commands
- Display configuration commands
- Statistics control commands
- Network management entities control commands
- Diagnostic control commands
- Change Network configuration commands

For many of these activities, you can use command files to simplify command entry. See chapter 3 for more information on command files.

Display Status Commands

Display status commands display the operational status of the hardware devices, communications lines, network solutions, and software configured for a DI or an ICA system.

A status display is similar to a snapshot in that it gives a picture of how the network is running at the time that the status command is processed. You can request and receive status displays anytime the network is running. They show you how the network component is performing at the time you request the status.

This snapshot effect is important when you are investigating user complaints or problems with the network. In such situations, you need to isolate the problem and return the user to network services as quickly as possible. Checking network component status is a first step in this process.

Displaying Network Components Status

In this activity, you request and obtain the current operational status of network components, such as hardware boards in a DI. You may route the displays to a file using SCL commands on NOS/VE or by the ROUTE_COMMAND_RESPONSE session control command on NOS.

To display the hardware status in your network with a logical system name of TDI_1, enter:

```
send_command command='display_hardware_status',s=tdi_1
```

You receive a response similar to the following:

```
Hardware Status
                                                                   boot ROM
device name  state   status         version    lim/bank/port  type  enab level
$MPB0        on      active         0000(16)                        n/a  160A
$PMM1        on      active         0008(16)                        n/a  160A
$SMM2        on      active         0001(16)   2                    n/a  0000
$CIM4        on      configured     0001(16)   0,1,2,3              no   2702
$CIM5        down    not config.    0000(16)                        yes  2702
$ESCI6       on      active         0000(16)                        no   0806

$LIM0        on      configured     0008(16)   4              RS232
$LIM1        down    configured     0009(16)   4              RS232
$LIM2        on      not config.    0000(16)   2              RS449
$LIM3        on      not config.    0000(16)   2              RS449
$URI4        on      configured
```

Communications Control Commands

Communications control commands start or stop communications on communications trunks; networks; and asynchronous, synchronous, or URI lines. The communications control commands address trunks, networks, and lines by the logical names assigned by define commands. These activities involve controlling the communications traffic on each specific communication line. Before performing these activities, make sure you know the network's physical configuration and the logical names assigned to the network's communication lines.

Starting and stopping lines may be done for several reasons, such as replacing a communication line and changing a line's logical configuration. Stopping a communication line cuts off a terminal user from the rest of the network. If you have to stop a line connected to a terminal, inform the terminal user well in advance that the line will be stopped by sending a WRITE_TERMINAL_MESSAGE command to the terminal user.

Starting a Line

To start an individual line, you must know the line and the logical names of the DI supporting the line. You may use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines.

Requirements:

- The line must be defined in the network's configuration by the DEFINE_LINE (DEFL) command. (A configured line is a line that has been assigned to a specific terminal interface program (TIP) that services the line when the line is started. If the line is not configured, a TIP has not been assigned to start and service the line.) If you're not sure the line is configured, check the DI's configuration file or enter the DISPLAY_LINE_STATUS command. Configured lines are indicated by `configured` in the command response.

- The terminal interface program (TIP) supporting this line must be configured by the DEFINE_TIP (DEFT) command. Check the DI's configuration file for this command.

  To start a line, enter the START_LINE command as shown in the following example.

  ```
  send_command command='start_line line_name=group_1',system=first_floor_tdi
  ```

Stopping a Line

To stop an individual line, you must know the logical name of the line. You can use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines. Use the following procedure.

1. Notify the line's user that the line will be stopped using the WRITE_TERMINAL_MESSAGE command. Tell user to log off.

   ```
   send_command command='write_terminal_message,ln=line23,..
   m=(''Line 23 going down please log off'')',s=tdi
   ```

2. Enter the STOP_LINE command as shown in the following example.

   ```
   send_command command='stop_line line_name=line23',system=tdi
   ```

Starting a Network Solution

Starting the network solution also starts the underlying trunk, if not already started.

Requirements:

- The network solution must be defined by the appropriate network definition commands. See Adding a Network Solution and Deleting a Network Solution later in this chapter.
- Know the network solution's logical name as it is defined for the DI to which you are sending the commands. Use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines.

Enter the START_NETWORK command as shown in the following example.

```
send_command command='start_network network_name=net_1',system=tdi04
```

Stopping a Network Solution

This activity affects a larger part of the CDCNET network than starting and stopping communication lines. Stopping a network solution logically removes a portion of the CDCNET network over which data can travel.

Do not stop the network solution that connects the operations station to the network host computer. Stopping the network solution which connects to the TDI that supports the operations station leaves the TDI (and you) logically disconnected from the network.

For example, if a TDI is connected to a CDCNET over a single Ethernet network solution, you should not stop communications on that network solution, because it is required to carry operations commands and other data to the TDI. You cannot start the network solution again unless you manually reset the TDI.

Requirements:

- Check the network's physical and logical configuration to determine the connections between DIs and network solutions. Do not stop a network solution if it is the only network solution over which your commands can be sent to a DI.
- Know the network solution's logical name. You may use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines.

Enter the STOP_NETWORK command as shown in the following example. The STOP_NETWORK command stops the underlying trunk if the network solution is the only traffic being carried by the trunk.

```
send_command command='stop_network network_name=net_1',system=tdi04
```

Starting Communications Trunks

To start the trunk, execute the START_TRUNK (STAT) command as shown in the following example.

```
senc c='start_trunk trunk_name = menlo_trunk_1' s=tdi2
```

Stopping Communications Trunks

To stop the communications trunk, execute the STOP_TRUNK (STOT) command as shown in the following example.

```
senc c='stop_trunk trunk_name = menlo_trunk_1' s=tdi2
```

Operator Messages Commands

Operator messages commands let you communicate with other operators on a communication path.

Sending Messages to Terminal Users

Sending the WRITE_TERMINAL_MESSAGE command through NETOU allows you to send messages to all users connected to a particular service, or to a particular line. You enclose the message within quotation marks. The optional parameters LINE_NAME, DEVICE_NAME, and SERVICE_NAME allow you to specify where you want the message to go.

You may send a message to a specific line or group of lines, to a particular terminal device or group of devices, or to the users of a specific gateway service. For example, if you send a message specifying a particular NOS/VE or NOS service name with the SERVICE_NAME parameter, all terminal users currently connected to the service name specified receives the message. Only terminals that match the parameters you specify receive this message. If you do not specify the optional parameters, the message is sent to all terminal users.

The message you specify with the message parameter must be entered as a string value enclosed by two consecutive apostrophes. If you want the message to have several lines of text, you must enter each line to be output at the terminal as a string value within parentheses, as in the following example.

The following command sends a message to a terminal user connected to TDI1 and on a line called LINE15:

```
send_command c='write_terminal_message,..
message=(''New communications configuration tomorrow'',''Network down ..
until 10:00.''),line_name=line15',system=tdi1
```

Receiving Messages from Terminal Users

Messages from terminal users are sent to the network operator by a terminal user command called REQUEST_NETWORK_OPERATOR (REQNO). These messages show up at your operations station as alarms. On NOS, a warning bell rings at an interactive terminal, and NETOU is displayed on the operator attention line at the host console.

The alarm message from a terminal user gives the line name, terminal device name, gateway service through which the message was sent, and the text of the message. You can route terminal user messages to a file using standard SCL commands on NOS/VE or the ROUTE_ALARM command on NOS.

Send a message back to the user using the WRITE_TERMINAL_MESSAGE command to acknowledge that you have received the message.

Example:

```
****** ALARM FROM  riverside_tdi_1              85/06/13 11.15.45 168
Terminal User Request
line_name = mech_eng_2
Device name = mech_eng_term_2
Message: Will be moving office next week.  Need configuration change form.
```

NOTE

The REQNO command does not execute successfully (a terminal user cannot contact the network operator using this command) unless CDCNET log message number 168 is enabled as an alarm by the DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM) configuration command on the DI. Message number 168 is not enabled as an alarm by default.

Clock Management Commands

Each CDCNET system reports date and time in command responses, logs, and alarms. So that the responses, logs, and alarms from different systems can be correlated, CDCNET provides clock management functions to ensure that all systems in a catenet report the same date and time (within 1 second) at the same instant. These functions are provided by the Independent and Dependent Clock MEs.

The Independent Clock ME resides in one system in a catenet and maintains the master clock for the catenet. A Dependent Clock ME resides in every system in a catenet. Each Dependent Clock ME is responsible for obtaining the master catenet clock from the Independent Clock ME and for setting the system's clock to the master clock.

If the Independent Clock ME resides in a CYBER 170 NOS MDI, you may reset the master clock through the SET_DATE_AND_TIME command. Through the SYNCHRONIZE_CLOCK command, which should be broadcast to each system in the catenet whenever the master clock is changed, you may reset each system's clock to the master clock.

There are three parts to the clock management function.

- Resetting the master clock, using the SET_DATE_AND_TIME command (NOS only).
- Synchronizing time clocks in all DIs, using the SYNCHRONIZE_CLOCK command.
- Displaying date and time set at a DI, using the DISPLAY_DATE_AND_TIME command.

NOTE

For CDCNET networks supported by a NOS/VE host, the master clock is configured in the NOS/VE host. For CDCNET networks supported by a NOS host, the master clock is configured in a DI in the network. This DI is called the clocking system DI.

Resetting the Master Clock (NOS Only)

1. Determine which DI contains the master clock by one of the following methods.

   - Check your site's records and network map (if available) for the DI marked as containing the master clock.

   - Send a DISPLAY_SYSTEM_OPTIONS (DISSO) command to each DI, specifying CLOCKING_SYSTEM with the DISPLAY_OPTION parameter.

     ```
     senc c='disso do=clocking_system',system=mdi_1
     ```

     The DI that contains the master clock sends the following response.

     ```
     clocking system = yes
     ```

2. Once you have located the DI containing the master clock, reset the master clock by sending a SET_DATE_AND_TIME command to that DI. Provide the current date and time for the DATE and TIME parameters. Both the date and time must be entered as string values enclosed by two consecutive apostrophes.

   In the following example, the master clock for a network is located in a DI called TDI2. To reset the master clock, the operator sends a SET_DATE_AND_TIME command to TDI2.

   ```
   senc c='setdat d=''11/24/85'',t=''08:25:49''',s=tdi2
   ```

After the master clock has been reset, synchronize all the DI clocks using the SYNCHRONIZE_CLOCK command, described next.

Synchronizing Time Clocks in All DIs

Clock synchronization automatically occurs when a DI is configured. Once a day, all DI clocks are resynchronized. Over one day's time, for example, clocks could be running 1 to 2 seconds out of synchronization with each other. The SYNCHRONIZE_CLOCK (SYNC) command synchronizes a DI's clock to the date and time set at the master clock.

If you want to synchronize the DI clocks, send the SYNCHRONIZE_CLOCK command to every DI in the network, or write and execute a command file that sends SYNCHRONIZE_CLOCK to every DI in the network (see chapter 3 for directions on writing a command file).

Displaying Date and Time Set at a DI

If, at any time, you want to see the date and time set at a DI, send the DI a DISPLAY_DATE_AND_TIME command as shown in the following example.

```
senc c='display_date_and_time',s=north_tdi_1

System date and time
    11/24/86 08:25:49
```

Display Configuration Commands

A display configuration command is provided for each network definition command. The display configuration commands display the current values of DI configuration parameters defined through network definition commands. These configuration parameters include the configuration of DI system software, hardware devices, communications lines, URI lines, network solutions, and interfaces.

Display Ethernet Trunk Configuration

One of the display configuration commands allows you to display the configuration of Ethernet trunks. The following example displays the configuration of the Ethernet trunk named ethernet_cim02.

To display the configuration of the selected Ethernet trunk, enter the following command.

```
senc c='display_ether_trunk_options tn=ethernet_cim02' s=tdi1
```

If the command executes successfully, you receive a response similar to the following:

```
ETHERNET trunk options
slot = 4
trunk_name = ETHERNET_cim02
max_frame_size = 1500
interframe_spacing = 96
```

Statistics Control Commands

Statistics control commands start and stop the collection of statistics for communications networks, lines, communications software, and trunks. In addition, the collection of statistics can be synchronized so that the statistic data reported by a system for different networks, lines, communications software, or trunks can be correlated.

The CDCNET statistics are counts of data traffic and various events detected by the communications software. The communications software gathers these statistics over a collection period called the report interval. The report interval is set through the start statistics commands and may differ between statistics. At the end of a report interval, the communications software reports the statistics via a log message, clears the collection counts and starts another report interval.

The three levels of statistics are summary, expanded, and debug. Summary level statistics provide an overview of the operation of a line, network solution, process, or trunk. The expanded and debug statistics provide further refinement of the statistics with the debug statistics the most detailed (some statistics do not support the expanded and debug levels).

Start Network Statistics

The following example starts statistics collection for a network named bld_3_net.

To start statistics collection, enter the following command:

```
senc c='start_network_metrics nn = bld_3_net' s=tdi3
```

If the command executes successfully, you receive a response similar to the following:

```
Network bld_3_net summary metrics started
```

Stop Network Statistics

To stop the collection of network statistics for one or more network solutions, enter the following command:

```
senc c='stop_network_metrics nn = bld_3_net,g = summary' s=tdi1
```

If the command executes successfully, you receive a response similar to the following:

```
Network BLD_3_NET summary stopped.
```

Network Management Entities Control Commands

These commands control the services provided by the following network management entities installed on CYBER 170 NOS MDI/MTI's.

- Operator Support Application
- Independent File Access ME
- **Initialization ME**

Cancel Operator Support Application

To cancel Operator Support Application for a NOS host, execute the CANCEL_OPERATOR_SUPPORT command as shown in the following example.

```
senc c='cancel_operator_support trunk_name = c170_trunk1' s=tdi4
```

If the command executes successfully, you receive a response similar to the following:

```
Operator Support is cancelled for trunk c170_trunk1
```

Diagnostic Control Commands

Diagnostic control commands place physical devices under diagnostic control and start and stop diagnostics on these devices. For detailed descriptions of the diagnostic commands, see the CDCNET Installation and Troubleshooting Guide.

Starting a Port Test

To start a diagnostic test on a given port, enter the following commmand:

```
senc c='start_port_test device_name = $lim_port1' s=tdi2
```

If the command executes successfully, you receive a response similar to the following:

```
PORT test started version 10
CIM slot number= 5
LIM slot number= 3
PORT number= 1
```

Change Network Configuration Commands

These commands change the configuration of communications trunks, networks, and lines.

Changing the Outcall Gateway

To change the outcall gateway default inactivity timer value, enter the following command:

```
senc c='change_outcall_gateway it = 30' s=td12
```

If the command executes successfully, you receive a response similar to the following:

```
Change of Outcall Gateway is accepted.
```

Advanced Activities

You need a deeper understanding of the network, its configuration, and software that runs in DIs and on host computers to perform the following advanced activities. Advanced activities include procedures that may affect the performance of the network, such as cancelling the logical configuration of a communication line. Such activities are usually performed by an analyst or by an operator under an analyst's supervision.

- Controlling the network services access.
- Changing the network's logical configuration.
- Controlling gateways.
- Logging and alarm control.
- Terminating network log files on NOS/VE.
- Terminating network log files on NOS.
- Archiving network log files.
- Loading and unloading software.
- Controlling the network delay measurement.

For many of these activities, you can use command files to simplify command entry. See chapter 3 for more information on command files.

Controlling the Network Services Access

CDCNET provides a Service Access Control feature to control network services that can be accessed across specific network solutions and DIs. Service Access Control operates by restricting access to service titles in the network directory. Users normally find services by requesting them by title. The title is translated to a network address by the directory management entity and a connection is established. Preventing a user from accessing a title also prevents that user from establishing a connection to the service.

Service Access Control features and procedures are further defined in the CDCNET Configuration Guide. The Service Access Control commands are defined in the CDCNET Commands manual.

Changing the Network's Logical Configuration

The activities in this subsection alter the logical configuration of the network using network operations commands to logically add, delete, and redefine communication lines, network solutions, gateways, log messages, and alarm messages.

There are several types of configuration changes. Some changes, such as the addition of new DIs and network solutions, can affect the entire network and its physical appearance. Other configuration changes are less visible, but are still physical changes, such as adding more lines to a DI.

Logical configuration changes are changes in the network's software, such as removing a network solution's definition from a DI's logical configuration, or changing the line speed and other attributes for a communication line. These changes are not as visible, but are no less important in affecting how the network operates.

Deleting an element from a network's logical configuration is as major a change as physically removing the element. A logically cancelled element can no longer be used to send, receive, or relay data. As a network operator, you may be called upon to change a DI's logical configuration. There are two ways to change a DI's logical configuration.

- Entering configuration commands through NETOU while the network is running.

  The same commands that are in a DI's configuration procedure (except the DEFINE_SYSTEM command) may be entered during operations. These commands change the logical configuration of the DI to which you send the command. This section assumes you are making configuration changes while the network is running.

  This type of configuration change made by entering commands through NETOU is not permanent. The configuration change at a DI stays in effect until that DI is reloaded. The configuration procedures on the host remain unchanged. When the DI is reloaded, the original configuration procedures are loaded. If you want to make permanent changes to a DI's logical configuration, you must access the DI's configuration procedures and make the changes to the procedures. You can use the MANAGE_CDCNET_CONFIGURATION (MANCC) Utility to edit configuration procedures.

- Changing the configuration by changing the configuration procedures.

  This type of change is more permanent because it stays in effect even if DIs are reloaded. However, these changes are permanent only if the system is reloaded. See the CDCNET Configuration Guide for information on MANCC.

Additional information on more advanced configuration changes such as changing terminal configuration parameters and reconfiguring a DI's base system software can be found in the CDCNET Configuration Guide.

Adding a Line

When a communication line is added to the network, it must be logically defined in addition to being physically installed. This definition consists of the line's logical name and characteristics of the line.

1. If a terminal interface program (TIP) has not been defined for the TDI or MTI supporting this line, define the TIP by the DEFINE_TIP command.

   ```
   send_command command='define_tip tip_type=asynctip',system=south_tdi_2
   ```

2. Define the line's configuration using the DEFINE_LINE command.

   ```
   send_command command='define_line lim=1,port=0,..
   tip_name=async,line_name=l10',system=south_tdi_2
   ```

3. The line should start after the DEFINE_LINE command completes, unless the optional START parameter was set to NO. If the line does not start communications, start the line (see Starting and Stopping Communication Lines in this chapter).

Deleting a Line

When communication lines are removed from the network, their definition must also be removed from the network's logical definition. To do this, enter a CANCEL_LINE command.

1. Notify user or users that the line or lines will be stopped using the WRITE_TERMINAL_MESSAGE command.

   ```
   senc c='write_terminal_message,ln=engin_line_31,..
   m=(''Line engin_line_31 being deleted'')',s=engin_tdi
   ```

2. Stop communications traffic on the line using the STOP_LINE command.

   ```
   senc c='stop_line line_name=engin_line_31',s=engin_tdi
   ```

3. Cancel the line's logical definition using the CANCEL_LINE command.

   ```
   senc c='cancel_line line_name=engin_line_31',s=engin_tdi
   ```

Redefining a Communication Line

To redefine a communication line, first cancel its current logical definition. Once the definition is cancelled, you can redefine the line using the DEFINE_LINE command. If the DEFINE_LINE command included the START=NO parameter, you must use the START_LINE command. Otherwise, the START_LINE command is unnecessary.

Enter the commands to redefine a line in the following sequence.

```
STOP_LINE
CANCEL_LINE
DEFINE_LINE
START_LINE
```

When network solutions are added to the network, they must be logically defined by configuration commands for the DIs using the network solutions. The configuration commands may be entered during operations, but changes remain in effect only until the DI is reloaded. To make permanent changes, the commands must be changed in the DI's configuration procedure (see the CDCNET Configuration Guide).

NOTE

Deleting the network solution over which you load the DI is not recommended.

Adding a Network Solution

When network solutions are added to the network, they must be logically defined by configuration commands for the DIs using the network solutions. The configuration commands may be entered during operations, but changes remain in effect only until the DI is reloaded. To make permanent changes, the commands must be changed in the DI's configuration procedure (see the CDCNET Configuration Guide).

Adding a network solution to a network's logical configuration involves defining the trunk which supports the network solution, then defining the network solution. The commands used for this depend on what type of network solution you are defining.

NOTE

When adding or changing a network solution, be sure to define the Service Access Control restrictions. See the CDCNET Configuration Guide for these procedures. The commands used to define or change Service Access Control are described in the CDCNET Commands manual.

- Ethernet Network Solutions

  For DIs loaded across an Ethernet medium (such as a TDI), the commands used to define an Ethernet trunk and network solution, DEFINE_ETHER_TRUNK and DEFINE_ETHER_NET, are performed implicitly by each DI's load process, and default names are assigned to the trunk and network solution. Once a DI is loaded and configured, you do not have to enter these commands through NETOU to define the Ethernet trunk and network solution. A DEFINE_ETHER_TRUNK or DEFINE_ETHER_NET command sent to such a DI fails if the trunk or network is already defined.

1. Enter the DEFINE_ETHER_TRUNK command. Provide the number of the slot in the DI which houses the ESCI board that supports the Ethernet trunk. If the DI has only one ESCI board, the slot number for the Ethernet trunk is optional. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the SLOT parameter, as in $ESCI4 (ESCI board in board slot 4).

```
senc c='define_ether_trunk trunk_name=ether1,slot=4',s=mdi_2
```

2. Enter the DEFINE_ETHER_NET command. Provide the logical names of the network solution and trunk, and the ID number assigned to the network solution. The trunk name must be the same as the trunk name specified in the DEFINE_ETHER_TRUNK command for the trunk to be used as a network solution.

```
senc c='define_ether_net network_name=ARHNET,trunk_name=ether1,..
network_id=0afbb1(16)',s=mdi_2
```

3. Enter the START_NETWORK command. Provide the logical name of the network assigned to the network by a define command.

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_ETHER_NET command.

- HDLC Network Solutions

1. Enter the DEFINE_HDLC_TRUNK command. Provide the numbers of the LIM and port to which the HDLC line is connected and which support the HDLC trunk. Provide the address of the local HDLC station and the address of the remote HDLC station. Both addresses are specified in digits from 0 through 9. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the LIM and PORT parameters, as in $LIM1_PORT3.

```
senc c='define_hdlc_trunk lim=1 port=1 local_address=3075551212,..
remote_address=5006221313 trunk_name=TYMN1' s=ndi_1
```

2. Enter the DEFINE_HDLC_NET command. Provide the trunk name, which must be the same as that specified on the DEFINE_HDLC_TRUNK command. Provide the network ID, which is the CDCNET network identifier of the HDLC network solution.

```
senc c='define_hdlc_network trunk_name=TYMN1,..
network_id=1234'..s=ndi_1
```

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_HDLC_NET command.

- X.25 Network Solutions

1. Enter the DEFINE_X25_TRUNK command. Provide the numbers of the LIM and port to which the X.25 line is connected, and which supports the X.25 trunk. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the LIM and PORT parameters, as in $LIM3_PORT1.

```
senc c='define_x25_trunk lim=1 port=1 trunk_name=TYMN1' s=ndi_1
```

2. Enter the DEFINE_X25_INTERFACE command. The trunk name must be the same as that specified on the DEFINE_X25_TRUNK command. The INONLY_RANGE, TWOWAY_RANGE, and OUTONLY_RANGE parameters specify ranges of channel numbers allotted for incoming calls and outgoing calls. At least one of these parameters must be specified. If you specify more than one range, the ranges must be ascending with no overlapping value ranges.

```
senc c='define_x25_interface trunk_name=TYMN1 public_data_network=TYMNET,..
twoway_range=0..32' s=ndi_1
```

3. Enter the DEFINE_X25_NET command. The trunk name is the name of the X.25 trunk that supports the network solution. The remote DTE address is the remote data terminating equipment address for this X.25 network solution. This is typically a telephone number for the other end of the network, assigned by the network provider (such as

   or Tymnet) when a site subscribes to the public data network. The address is specified as a string of digits from 0 through 9. The network ID is the CDCNET network identifier of the X.25 network solution.

```
senc c='define_x25_network trunk_name=TYMN1,..
remote_dte_address=''3075551212'' network_name=TYMNET_NET1,..
network_id=1234' s=ndi_1
```

4. If the X.25 network solution connects to foreign hosts, you must enter a DEFINE_X25_GW command to define the gateway between CDCNET and the foreign host.

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_X25_NET command.

Deleting a Network Solution

A network solution can be logically deleted. However, the network solution should not be deleted if it is the only link between a DI and the rest of the network. For example, if you logically delete the Ethernet network solution which is the only path from a TDI to the rest of the network, you cut off that TDI from the rest of the network. You cannot access the TDI by NETOU to reenable the network solution; the only way to redefine the network solution is to manually reset the TDI.

- Ethernet Network Solutions

  To delete an Ethernet network solution, follow this procedure.

  1. Stop traffic on the network solution by entering the STOP_NETWORK command.

     ```
     send_command c='stop_network network_name=engin_bldg_net',s=engin_tdi_1
     ```

  2. Cancel the network solution's definition by entering the CANCEL_ETHER_NET command. This command also cancels the underlying Ethernet trunk, so a separate CANCEL_ETHER_TRUNK command is not needed. However, when redefining an Ethernet network solution, you must define the Ethernet trunk because it was cancelled (see Redefining a Network Solution in this chapter). Provide the logical name of the network solution for the NETWORK_NAME parameter.

     ```
     send_command c='cancel_ether_net network_name=engin_bldg_net',s=engin_tdi_1
     ```

- NP Interface (NOS Only)

  To logically delete an NP interface on NOS, follow this procedure.

  1. Stop traffic to a NOS Network Products host by entering the STOP_NP_INTERFACE command. This command identifies the NOS Network Products interface to the NOS host. Provide the logical name of the interface assigned by the DEFINE_NP_INTERFACE configuration command for the INTERFACE_NAME parameter.

     ```
     senc c='stop_np_interface in=cyber_109', s=mdi1
     ```

  2. Cancel the configuration of the NP interface with a CANCEL_NP_INTERFACE command. Provide the logical name of the interface assigned by the configuration command, DEFINE_NP_INTERFACE for the INTERFACE_NAME parameter.

     ```
     senc c='cancel_np_interface in=cyber_109', s=mdi1
     ```

  3. Cancel the configuration of the channel trunk with a CANCEL_CHANNEL_TRUNK command. Provide the logical name of the trunk assigned by the configuration command, DEFINE_CHANNEL_TRUNK for the TRUNK_NAME parameter.

     ```
     senc c='cancel_channel_trunk tn=cyber_101_alt'
     ```

- HDLC Network Solutions

  To logically delete an HDLC network solution, follow this procedure.

  1. Stop traffic on the network solution by entering the STOP_NETWORK command. Provide the logical name of the network solution for the NETWORK_NAME parameter.

     ```
     send_command c='stop_network network_name=tymnet_net_1',s=ndi_1
     ```

  2. Cancel the HDLC network solution by cancelling the logical definition of the HDLC network and the HDLC trunk by entering the CANCEL_HDLC_NET command. This also cancels the underlying trunk definition. Provide the logical name of the HDLC network for the NETWORK_NAME parameter.

     ```
     send_command c='cancel_hdlc_net network_name=tymnet_net_1',s=ndi_1
     ```

- X.25 Network Solutions

  To logically delete an X.25 network solution, follow this procedure.

  1. Stop traffic on the network solution by entering the STOP_NETWORK command. Provide the logical name of the network solution for the NETWORK_NAME parameter.

     ```
     send_command c='stop_network network_name=tymnet_net_1',s=ndi_1
     ```

  2. Cancel the network solution's definition by entering the CANCEL_X25_NET command. Provide the logical name of the network solution for the NETWORK_NAME parameter.

     ```
     send_command c='cancel_x25_net network_name=tymnet_net_1',s=ndi_1
     ```

  3. Stop the X.25 Packet Level interface by entering the STOP_X25_INTERFACE command. Provide the logical name of the X.25 interface for the INTERFACE_NAME parameter.

     ```
     send_command c='stop_x25_interface network_name=tymnet_net_1',s=ndi_1
     ```

  4. If the X.25 interface that supports the network solution is also to be cancelled, enter the CANCEL_X25_INTERFACE command. If the X.25 interface has other active users, such as an X.25 gateway, do not cancel the X.25 interface. Provide the logical name of the interface assigned by a DEFINE_X25_INTERFACE configuration command for the INTERFACE_NAME parameter.

     ```
     send_command c='cancel_x25_interface interface_name=tymnet_1',s=ndi_1
     ```

  5. If the logical definition of the trunk that supports the network solution is also to be cancelled, enter the CANCEL_X25_TRUNK command. Provide the logical name of the trunk for the TRUNK_NAME parameter.

     If the X.25 interface remains, do not cancel the trunk.

     ```
     send_command c='cancel_x25_trunk trunk_name=tymnet_trunk_1',s=ndi_1
     ```

Redefining a Network Solution

To redefine a network solution's logical definition, first cancel the current definition, then provide the values for the new definition. This subsection presents the sequence of commands required to redefine Ethernet, channel, X.25, and HDLC network solutions.

- Ethernet

  The CANCEL_ETHER_NET also cancels the underlying Ethernet trunk, so a separate CANCEL_ETHER_TRUNK command is not needed. However, when redefining an Ethernet network solution, you have to define the Ethernet trunk, since it was cancelled.

  STOP_NETWORK
  CANCEL_ETHER_NET
  DEFINE_ETHER_TRUNK
  DEFINE_ETHER_NET
  START_NETWORK

- Channel (NOS Only)

  STOP_NP_INTERFACE
  CANCEL_NP_INTERFACE
  CANCEL_CHANNEL_TRUNK
  DEFINE_CHANNEL_TRUNK
  DEFINE_NP_INTERFACE

- X.25

  STOP_NETWORK
  CANCEL_X25_NET
  STOP_X25_INTERFACE
  CANCEL_X25_INTERFACE
  CANCEL_X25_GW (if applicable)
  CANCEL_X25_TRUNK
  DEFINE_X25_TRUNK
  DEFINE_X25_INTERFACE
  DEFINE_X25_NET
  DEFINE_X25_GW (if applicable)
  START_NETWORK

  If you only want to redefine the network solution, enter the following commands.

  STOP_NETWORK
  CANCEL_X25_NET
  DEFINE_X25_NET
  START_NETWORK

- HDLC

  STOP_NETWORK
  CANCEL_HDLC_TRUNK
  DEFINE_HDLC_TRUNK
  DEFINE_HDLC_NET
  START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once it is configured. (By default, it is started.) This is set by the START parameter on the DEFINE_ETHER_NET, DEFINE_X25_NET, and DEFINE_HDLC_NET commands.

Adding Terminal Devices

To add a terminal device to a DI's logical configuration, create a terminal definition procedure (TDP) that contains the DEFINE_TERMINAL_DEVICE command. See the CDCNET Configuration Guide for information about creating TDPs.

Adding Batch Devices, I/O Stations, and NTF Remote Systems

Logical configuration of batch devices, I/O stations, and NTF remote systems is covered in the CDCNET Configuration Guide. Operation of batch devices is covered in the CDCNET Batch Device User Guide and the NOS Remote Batch Facility (RBF) Reference manual. See these manuals for detailed information on configuring and operating batch devices.

To configure batch I/O stations and individual devices, use TDPs. TDPs contain commands to define the logical group of batch devices called an I/O station, to define parameters that apply to all the devices in the I/O station, and to define parameters that apply to the individual batch devices such as printers in the I/O station. The following commands are used in TDPs for I/O stations.

DEFINE_BATCH_DEVICE
DEFINE_I_O_STATION
DEFINE_NP_BATCH_STATION
DEFINE_TERMINAL_DEVICE
DEFINE_USER_I_O_STATION

Network Transfer Facility (NTF) Remote Systems are configured using TDPs. The following commands are used in TDPs for NTF Remote Systems:

DEFINE_ACCESSIBLE_REMOTE_SYSTEM
DEFINE_BATCH_STREAM
DEFINE_REMOTE_SYSTEM

TDPs are created during network configuration, but they can be modified and new ones can be created. See the CDCNET Configuration Guide for more information on creating and modifying TDPs and configuring I/O stations and NTF remote systems. TDPs are either executed automatically when the line connected to the I/O or remote system station becomes active, or when a station operator executes the TDP using the DO command. For example, the following command executes a TDP named STATION1. PROCEDURE_TYPE=TDP is required.

```
DO,STATION1 PT=TDP
```

If you are already connected to a host service, use the network command character with the DO command, as shown in the following example.

```
%DO,STATION1 PT=TDP
```

Once batch I/O stations and their devices are active, you can perform operations such as starting and stopping devices as described in the CDCNET Batch Device User Guide and the NOS RBF Reference manual.

Controlling Gateways

This section describes how to control gateways.

Network Products Gateways

Activities are usually done by including the commands in the DI configuration files. The DEFINE_NP_GW command automatically starts the gateway when the command executes, so a start command is not currently supported. There are commands to start and cancel the Network Products interface (START_NP_INTERFACE and CANCEL_NP_INTERFACE).

The ADD_NP_GW_OUTCALL is used when a remote system must access applications residing on a NOS host. The outcall is from the perspective of the CDCNET network; the call is going out of the CDCNET network. The add command provides the name (title) of the application through which remote systems can access applications residing on a NOS host. The name (title) is registered and maintained on a directory by the Directory Management Entity.

X.25 Gateways

The following commands are used to control access to foreign hosts connected to X.25 networks:

START_X25_INTERFACE
STOP_X25_INTERFACE
CANCEL_X25_INTERFACE
DEFINE_X25_INTERFACE
START_X25_GW
STOP_X25_GW
CANCEL_X25_GW
DEFINE_X25_GW
ADD_X25_GW_OUTCALL
DELETE_X25_GW_OUTCALL
DEFINE_X25_TERMINAL_GW
START_X25_TERMINAL_GW
STOP_X25_TERMINAL_GW
CANCEL_X25_TERMINAL_GW

The start, stop, cancel, and define commands control the X.25 interface. The start, stop, cancel, and define X.25 gateway (GW) commands control the X.25 gateway that provides access for NOS applications-to-applications on foreign systems connected to CDCNET by an X.25 public data network. The add and delete commands control the registration of the name (title) of the X.25 gateway in the Directory ME. The X.25 interface supports the X.25 gateway. When starting the X.25 gateway, first start the interface. When stopping the interface, you must first stop the X.25 gateway, and if an X.25 network solution is defined, the X.25 network solution. Stopping the X.25 interface also stops the trunk that supports the interface.

Figure 4-1 shows how the X.25 control commands are used to start and stop X.25 gateway services.

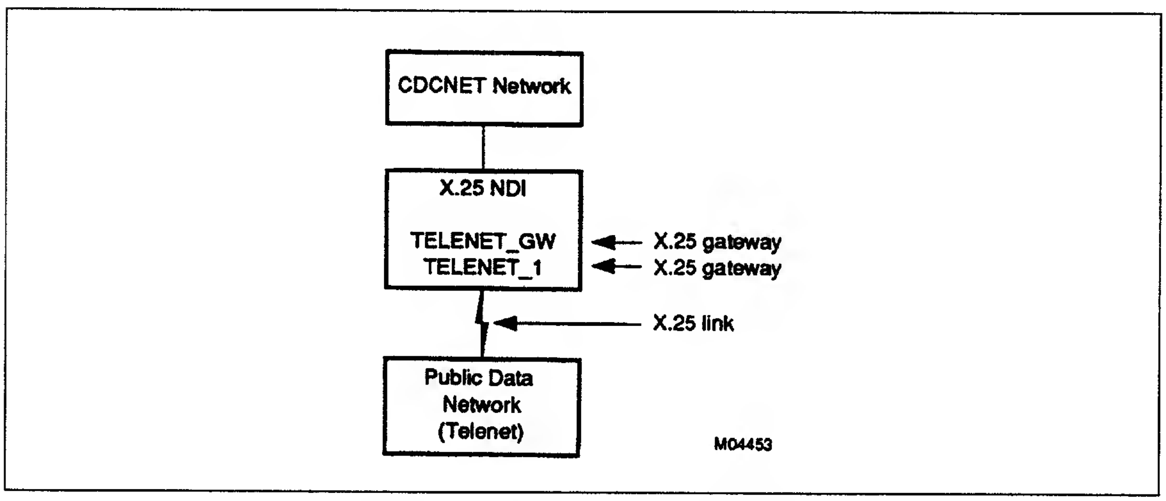


Figure 4-1. X.25 Gateway Example

Figure 4-1 also shows an NDI connecting a CDCNET network with TELENET, a public data network, over an X.25 link. The X.25 interface to TELENET was defined during configuration in the NDI by the DEFINE_X25_INTERFACE, ADD_X25_GATEWAY_OUTCALL, and DEFINE_X25_GW commands. The logical name for the X.25 interface is TELENET_1. The logical name for the X.25 gateway is TELENET_GW. CDCNET terminal users can access TELENET by starting and stopping X.25 gateway, TELENET_GW.

To start X.25 gateway services, the following commands are sent to the NDI.

```
senc c='start_x25_interface interface_name=telenet_1' s=ndi_1

senc c='start_x25_gateway gateway_name=telenet_gw' s=ndi_1

senc c='add_x25_gateway_outcall gateway_name=telenet_gw title=PTFS$' s=ndi_1
```

To stop X.25 gateway services, the following commands are sent to the NDI. The stop commands are sent in the opposite order of the start commands.

```
senc c='delete_x25_gateway_outcall gateway_name=telenet_gw title=PTFS$' s=ndi_1

senc c='stop_x25_gateway gateway_name=telenet_gw' s=ndi_1

senc c='stop_x25_interface interface_name=telenet_1' s=ndi_1
```

TCP/IP Gateways

Commands can define and start TCP/IP gateways, but these activities are usually done by including the commands in the DI configuration files. The DEFINE_USER_TELNET_GW command automatically starts the user gateway when the command executes. The DEFINE_SERVER_TELNET_GW command automatically starts the server gateway when the command executes. Include a DEFINE_IP_HOST command for the CYBER host.

The following examples illustrate how to cancel and redefine USER_TELNET and SERVER_TELNET gateways. In the first example, the IP_ADDRESS parameter of the DEFINE_USER_TELNET_GW command is the IP_ADDRESS of a non-CYBER host providing interactive services on the TCP/IP network.

In the second example, the IP_ADDRESS parameter of the DEFINE_SERVER_TELNET_GW command must match the IP_ADDRESS of the DEFINE_IP_HOST command that configured the CYBER host providing the interactive services on the TCP/IP network.

```
senc c='stop_user_telnet_gw gateway_name=gw_to_vax',s=ndi_1
senc c='cancel_user_telnet_gw gateway_name=gw_to_vax',s=ndi_1
senc c='define_user_telnet_gw gateway_name=gw_to_vax,..
ip_address=(128,5,0,3),..
title=vax_86',s=ndi_1

senc c='stop_server_telnet_gw gateway_name=gw_to_cyber',s=ndi_3
senc c='cancel_server_telnet_gw gateway_name=gw_to_cyber',s=ndi_3
senc c='define_server_telnet_gw gateway_name=gw_to_cyber,..
ip_address=(128,5,0,2),..
title=VE_990',s=ndi_3
```

IP Host

Commands can cancel and redefine an IP host. The host must have been previously defined with the DEFINE_IP_HOST command.

The following example shows how to cancel and redefine the IP host.

```
senc c='cancel_ip_host ip_address=(128,5,0,3)'
senc c='define_ip_host ip_address=(128,5,0,3),..
                        host_type= ip_host,..
                        system_id=(070701(16),009ECB(16))'
```

Logging and Alarm Control

This section describes activities for configuring and managing the CDCNET log and alarm message features. Network logging allows you to have a record of network activity in the form of log messages routed to a file on the host computer. Alarms are messages sent to your operations station that alert you to events in the network.

This section also refers to the utility that terminates the network log file on a NOS host, the Network Logfile Termination (NLTERM) Utility. If you are running CDCNET with a NOS host, you have to use NLTERM periodically to close the current network log file and write the log messages to another permanent file.

Defining Log Messages To Be Generated by a DI

The CDCNET logging structure consists of log message sources and log message recorders. Each DI is a log message source. The source provides log messages that describe the DI's activities. Each log message has a unique log message identifier. The complete list of these log messages and their identifiers is in the online Diagnostic Messages manual.

In CDCNET networks connected to a NOS host, at least one DI in the network serves as a log message recorder. The recorder has access to permanent storage. Aided by a NOS CDCNET host application called the Network Log Server (NETLS), the recorder DI records the log messages from the source DIs into a host file known as the network log file. The network log file on NOS resides on family SYSTEMX. The log file name has the following format:

Value	Description
a	Character incremented for each new log file, as in A,B,C.
mmdd	Date file was created.

In CDCNET networks connected to a NOS/VE host, the log message recording function is configured in the NOS/VE host. A log message recorder DI cannot be defined in NOS/VE environments. Commands in the NOS/VE host $SYSTEM.PROLOGS_AND_EPILOGS.NETWORK_ACTIVATION_EPILOG file activate and deactivate the network logging function: ACTIVATE_NETWORK_LOG and DEACTIVATE_NETWORK_LOG. For more information on these commands, see the NOS/VE System Analyst Reference set, Network Management.

There are network operations commands to configure and reconfigure the logging structure of your network. At each DI, there are lists maintained of what messages should be logged. The commands that affect logging sources allow you to define, change, and cancel one or more log messages at each DI.

If you have logging sources defined in your network, you should have a logging recorder defined for the network, or a portion of the memory in the network's DIs are used up by queued log messages generated by the DIs.

During network configuration, a default set of log message numbers are defined for each DI in the network with the DEFINE_SOURCE_LOG_GROUP command. These default messages are defined by commands in the DI configuration files created by the site administrator. Information on this activity is provided in the CDCNET Configuration Guide. You may add messages to this default set, but it is not recommended that you delete messages from the default set.

You can use the Network Performance Analyzer (NPA) Utility to look at log messages (see chapter 5 for NPA information).

Adding Log Messages to the Currently Defined List for Source DIs

1. Display the log messages that are currently logged at the source DIs using the DISPLAY_SOURCE_LOG_GROUP command.

2. Add or delete the messages you want to enable or disable using the CHANGE_SOURCE_LOG_GROUP command. See the online CDCNET Diagnostic Messages manual for message numbers.

Cancelling and Redefining Log Messages

You can also cancel and redefine the list of log messages to be generated at a DI by using the CANCEL_SOURCE_LOG_GROUP and DEFINE_SOURCE_LOG_GROUP commands.

CAUTION

It is recommended that you limit the number of log messages generated by a DI, since the messages are logged on the host disk space. If a large number of messages, particularly the entire set of log messages, are enabled for a DI, a significant amount of network traffic is dedicated to transmitting log messages to the host. The log message feature may be useful for tracking problems or events in the network. However, enabling too many log messages can put constraints on DI and host memory.

Changing the Logging Recorder DI (NOS Only)

In this activity, you control which host is to record log messages. This procedure is performed only in CDCNET networks that are supported by NOS hosts. Address the commands only to MDIs/MTIs that provide for log recording.

To cancel and redefine the recorder log group, follow this procedure:

1. Cancel the current log group to be recorded using the CANCEL_RECORDER_LOG_GROUP command.

2. Redefine the log group to be recorded using the DEFINE_RECORDER_LOG_GROUP command.

   NOTE

   If you cancel the recorder log group, you cancel the recording function for the entire catenet unless the log recording function is defined on multiple MDIs in the catenet. Cancelling and redefining a log recording function should be done only if you move the log message recording function from one DI to another.

Alarm Control

During network configuration, a default set of alarm message numbers are defined for each DI in the network by the DEFINE_SOURCE_ALARM_MESSAGE command. These alarms are sent to an alarm recorder. For NOS/VE operating systems, this is a network operator that executes the ACTIVATE_ALARMS command. Information on this activity is provided in the CDCNET Configuration Guide. You may add messages to this default set, but it is not recommended that you delete messages from the default set.

The initial set of DIs that report alarm messages to your operations terminal or console is all the DIs in the catenet. NOS/VE requires you to enter the ACTIVATE_ALARMS command within NETOU in order to receive alarms from the DIs. NOS has no such requirement; the alarms activate by default on NOS.

Occasionally, you may choose to redefine the list of alarm messages and/or the set of DIs that report alarms to you. For example, if a DI is undergoing tests and generating many alarms, and the DI is being monitored by test personnel, you can shut off receipt of the alarms from that DI.

There are two main activities involved in alarm control.

- Defining alarm messages to be delivered from a DI.

  This activity allows you to add and delete alarm messages from the list of alarms which are to be reported to all operators in the network from a particular DI.

- Controlling your alarm environment (NOS Only)

  This activity allows you to control which DIs report alarms to you. You may temporarily shut off receipt of alarms from a DI at your operations terminal/console. See chapter 3 for commands which control your operations alarm environment.

Defining Alarm Messages To Be Generated by a DI

To initially define the set of alarm messages to be delivered from the source DI, use the DEFINE_SOURCE_ALARM_MESSAGE command. See the online CDCNET Diagnostic Messages manual for the message numbers.

CAUTION

It is recommended that you limit the number of alarm messages defined for a DI. If a large number of alarms are enabled, the amount of network traffic devoted to alarm message transmission is increased. In addition, your operations station is constantly receiving alarms. The alarm message feature may be useful for tracking problems or events in the network. However, enabling too many alarms can put constraints on available DI memory.

Controlling Alarm Environment

To redefine the set of messages to be delivered from the source DI as alarms, enter the following commands.

1. DISPLAY_SOURCE_ALARMS (to display alarm messages enabled).
2. CANCEL_SOURCE_ALARM_MESSAGE (to delete messages).
3. DEFINE_SOURCE_ALARM_MESSAGE (to add messages).

Provide the identification numbers for the messages you want the DI to send as alarms, surrounded by parentheses. See the online Diagnostic Messages manual for the message numbers. To add alarm messages to the existing set, you can enter a DEFINE_SOURCE_ALARM_MESSAGE without having to cancel the existing set of messages.

NOTE

In order for the REQUEST_NETWORK_OPERATOR (REQNO) terminal user command to work, message number 168 must be enabled as an alarm.

Terminating Network Log Files on NOS/VE

Control Data host computers provide logging capabilities to the network. Hosts maintain a network log file that receives log messages sent from DIs. Periodically, the current network log file must be terminated, and a new file to which new log messages are written must be defined.

On NOS/VE, the network log file resides on file LOG in the $SYSTEM.CDCNET catalog. Individual sites can define the log file size limit, maximum number of log file cycles, and the interval at which a log file is terminated and analyzed, by specifying these values as parameters on the ACTIVATE_NETWORK_LOG command. This command can be entered by a system operator or be included in the NOS/VE host's $SYSTEM.PROLOGS_AND_EPILOGS.NETWORK_ACTIVATION_EPILOG file. This file is executed when NAM/VE is started. For more information on the ACTIVATE_NETWORK_LOG command, see the NOS/VE System Analyst Reference Set, Network Management.

Parameters used to define log file termination and processing on the ACTIVATE_NETWORK_LOG command include MAXIMUM_LOG_CYCLES, MAXIMUM_LOG_SIZE, and INTERVAL. A site-managed job is used to process the log file.

MAXIMUM_LOG_CYCLES specifies the maximum number of log file cycles allowed. When this limit is reached, logging is suspended until one or more log file cycles are deleted. The default value is 999 cycles.

MAXIMUM_LOG_SIZE specifies the maximum size (in bytes) of the log file. When this file size is reached, the log file is terminated, a file called PROCESS_LOG_JOB is submitted as a batch job (see the following description of PROCESS_LOG_JOB), and a new log file cycle is started. The default maximum file size for log files is the NOS/VE maximum file size as configured for your site. If the keyword value NONE is specified for this parameter, the NOS/VE maximum file size is used.

INTERVAL establishes the time interval (in minutes) at which the log file is to be terminated and processed and a new log file created. The default for this parameter is for no periodic processing; a log file is terminated when it reaches a certain file size, rather than when a time period elapses.

The PROCESS_LOG_JOB is a batch job that is automatically run each time a network log file is terminated. The PROCESS_LOG_JOB resides in the file $SYSTEM.CDCNET.SITE_CONTROLLED.PROCESS_LOG_JOB. The sample job consists of three separate nested batch jobs. The three jobs execute NPA commands that reformat inactive cycles of the CDCNET log file, create and print NPA reports, and back up copies of processed log files, and archive old data in the NPA data bases. Control Data provides functioning versions of these jobs, which are written in System Command Language and are self-documenting. Examine the jobs and change them to meet the needs of your job site. The sample jobs may be found on file $SYSTEM.CDCNET.VERSION_INDEPENDENT.PROCESS_LOG_JOB.

Terminating Network Log Files on NOS

On NOS, the network log file is a NOS direct access permanent file under user name SYSTEMX. The network log file is not automatically terminated. You must use the Network Log File Termination Utility (NLTERM) to terminate log files. The Network Logfile List Utility (NLLIST) provides a list of all terminated network log files that have not been purged. The function of NLLIST is also performed by an NLTERM subcommand called LIST.

NLTERM can be run as part of a daily system closedown process submitted as a batch job.

Archiving Network Log Files

Archiving log files that have been terminated is an additional log file management step which may be appropriate for your site, depending on your site's network configuration, how much log traffic is generated, and how large your log files are. See chapter 5 for more information on archiving log files.

Measuring Network Delay

Network delay time can be measured against a user-specified delay-time threshold by sending messages one at a time and waiting for the message to return. The delay time is compared against the specified delay time to determine if the message exceeds the threshold.

At the completion of the measurement period, a log message is issued indicating the average delay time for all messages during the transmission period. If the error threshold is exceeded, a network alarm message is issued. You may optionally define or cancel any log messages and network alarms by using CDCNET network commands.

Starting Network Delay Measurement

To initiate the network delay measurement feature, execute the START_NET_DELAY_MEASUREMENT (STANDM) command. The STANDM command validates the input parameters, loads and starts the measurement task, and issues a command response indicating the start status of the command. Enter the STANDM command as shown in the following example.

```
senc 'start_net_delay_measurement destination_system=ndi_d2..
delay_time_threshold=400' s=mdi_a1
```

Stopping Network Delay Measurement

To stop the network delay measurement feature, enter the STOP_NET_DELAY_MEASUREMENT (STONDM) command as shown in the following example.

```
senc 'stop_net_delay_measurement destination_system=ndi_1' s=mdi_a1
```

Displaying Network Delay Measurement Results

To display the network delay measurement results of a running measurement, execute the DISPLAY_NET_DELAY_MEASUREMENT (DISNDM) command. This command displays the current measurement parameters and statistics as shown in the following example.

```
senc 'display_net_delay_measurement destination_system=ndi_d2..
      dtt=450 imi=15' s=ndi_a1

FROM NDI_A1
  Network Delay Measures:
  To NDI_1
    Delay_time_threshold (msec.) = 450
    Average_only = NO
    Measurements = CONTINUOUS
    Messages_per_measurement = 25
    Measurement_priority = INTERACTIVE
    Inter_measurement_interval (minutes) = 15
    Error_threshold = 1
    Message_timeout (seconds) = 120

    Measurements completed = 12
    Last delay time average (msec.) = 47
      at 13:43:22 on 01/16/90
```

Network Performance Analyzer (NPA) Functional Overview 5

Network Performance Analyzer (NPA) Functional Overview 5

This chapter provides a functional overview of how NPA generates the reports used to monitor and troubleshoot a network. See chapter 6 for examples of all NPA reports. The following topics are included in this chapter.

NOTE

For more detailed information on the commands discussed in this chapter, see the CDCNET Commands Reference manual.

- NPA Features
- Functional Overview
- How to Initiate NPA Reports
- How to Enter NPA Commands in Screen Mode Format (NOS Only)
- How to Enter NPA Commands in Line Mode Format
- How to Get Help on NPA Procedures

NOTE

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual. The CDCNET Network Management Station has a utility similar to NPA.

NPA Features

NPA consists of two major and four minor software components. The two major components are:

- Data Reformat
- Report Generator

The four minor components are:

- File Maintenance Utilities
- Help File Utilities
- Change Expected Operating Limits
- Periodic Utility (NOS only)

Data Reformat

The reformat process should be initiated during the operating system startup and then periodically scheduled. The function of the reformat process is to:

- Acquire network log file(s) from the host system on which NPA runs, and map that information into the data file formats. CDCNET, at this time, only acquires log files from the host on which NPA is running.
- Collect the statistical information and the event information into the database files.
- Provide an indication of when a log file has been completely processed. The log file may then be dumped and purged.

The reformat process can automatically find log files under a catalog or acquire a log file specified by the user.

On NOS, if a log file is not specified as an input parameter, then the reformat process automatically searches a specified user catalog for all files that begin with the two letters NL and the current active log file, NETLFmid (up to a maximum of 30 files). These files are acquired if not already local to the job. If a user catalog is not specified, then the user catalog SYSTEMX is used. Access to all files is totally controlled via standard NOS file permissions (including log files in SYSTEMX).

On NOS/VE, if a log file is not specified as an input parameter, then the reformatter accesses and processes all cycles of the $SYSTEM.CDCNET.LOG file, including the current (highest) cycle (up to a maximum of 30 cycles). Access to log files is controlled via standard NOS/VE file access permissions.

Network log files that have been previously processed by the reformatter may be reprocessed, but duplicate information remains in the database files. Report generation ignores these duplicate records. However, the cost to the user is extra file space and report generation time. Therefore, you should avoid the practice of reprocessing log files.

The reformatter processes each log message that has been acquired from the log files and maps it into the appropriate NPA database record. There are three types of NPA database records:

- Event records
- Statistics records
- Account records

Event records correspond one-to-one to a log message. The only difference is that they are reformatted from binary log format to readable text. An example of an event record is a software message record.

Statistics records correspond one-to-one to a log message. The only difference is that they are reformatted from binary log format into an internal fixed record NPA-defined format.

Account records are in log file format. Accounting log messages are copied as is to the accounting database. No reformatting occurs.

The date and time from the device interface (DI) clock are taken when the log record is reported to the Dependent Log Management Entity (Dependent Log ME). The time is universal network time, which can be either Greenwich Mean Time, if the network nodes span several time zones, or local time of the host for smaller localized networks.

It is necessary that a coordinated time base be used so that the time that hardware/software events occur in one DI corresponds to the time that other network events occur.

The reformat process writes the reformatted text output on a standard CYBER 170 or 180 sequential file(s).

5

Report Generator

The function of the report generator is to generate standard, predefined reports or groups of reports upon request. You specify, by keywords, individual report names or the name of a report series, oriented toward a given target audience. The databases used are selected by the command parameters that you use.

NPA uses a Control Data product known as Information Processing Family Version 2 (IPF2) as part of its internal system. If you are a fully licensed IPF2 user, you can interactively manipulate the NPA database files to lay out customized report formats and to generate reports that meet your specific needs.

This capability can be used for developing reports that you can add to the set of NPA-supported reports. However, creating specialized reports requires additional host central-processor and central-memory resources beyond those required for normal NPA reporting. This capability also requires the use of the COBOL compiler. Therefore, an additional product license may be required if your site does not already have the COBOL product.

Chapter 7 provides an example of how to create a customized report using the IPF2 database files. If you need more information on the IPF2 files, refer to the IPF2 Reference manual listed in Additional Related Manuals.

File Maintenance Utilities

These utilities consist of the following processes:

- Archiving databases
- Reloading databases

The archiving process is used to move older processed data to alternative storage media. This process allows you to release disk storage space. You determine how often and when to archive based on the amount of mass storage available at your site. The parameters for these procedures determine which data records are to be manipulated and the time interval during which the selected data was collected and recorded to the network log file.

The reloading process is used to reload records from an archive file and merge these records into the existing databases.

Help File Utilities

The help file utilities consist of the following two interactive processes:

- Explain log messages
- Edit log messages

The EXPLAIN_CDCNET_LOG_MESSAGES (EXPCLM) command provides log message information on the selected log message number.

The EDIT_CDCNET_LOG_MESSAGES (EDICLM) command allows you to edit the help information for the selected log message.

Change Expected Operating Limits

Some NPA reports contain the expected operating limits feature. This feature is a means of directing your attention to events that occur less frequently than or more often than expected for satisfactory network performance.

The CHANGE_EXPECTED_OPERATING_LIMITS (CHAEOL) command allows selection of a lower and an upper limit for columnar data within the report. When the number of times that an event occurs during a reporting period falls between the lower and upper CHAEOL limits, it is considered to be satisfactory. If an event occurs less frequently than expected, a less-than symbol (<) appears to the right of the reported statistic. If an event occurs more often than expected, a greater-than symbol (>) appears to the right of the reported statistic.

NOTE

On NOS, the CHAEOL command is only applicable in full screen mode.

Periodic Utility (NOS Only)

The NOS utility SUBBJP allows periodic submission of a job file to the input queue. SUBBJP determines if a file is ready to be submitted. After that file has been submitted, or if there is no file ready to be submitted, the utility rolls out until another file is submitted. A file is submitted if the difference between the last time it was scheduled and the current time exceeds the periodic interval specified. The last time executed is then updated. No provision is made for jobs that abnormally terminate.

Functional Overview

NPA consists of the following four functions:

- Data collection
- Data logging
- Data reformatting
- Data reporting

Data Collection

Software routines built into your DI units are responsible for the initial collection of appropriate network analysis data.

CDCNET is designed to collect data periodically. The normal report interval is 3600 seconds. This means that data is collected once every hour, giving you 24 sample measurements for each data field every day. Data is written to the appropriate log file as it is collected.

If you need sample measurements more often than hourly, you may reduce the report interval through the use of network operator commands defined in chapter 4. However, if you severely reduce the report interval, the amount of time your central processor unit (CPU) spends on statistics collection increases, and the amount of time and resources spent on processing work decreases. Therefore, a report interval of between 10 and 15 seconds is a practical lower limit.

Data collection software routines transform the collected data into log messages. There are four types of log messages collected:

Accounting information	Information that helps to determine the distribution of costs to the user (see Accounting Database Restrictions later in this chapter).
Event information	Information concerning hardware and software errors and other software events.
Statistics information	Information that is accumulated periodically on the operation of the network (see Statistics Control later in this chapter).
Configuration information	Each DI reports its current configuration to the log file once every hour. In addition, any configuration change is reported at the time of its occurrence.

Data Logging

The logging process consists of logging capabilities built into CDCNET software routines. A resident DI application task performs data logging at preset intervals. Statistics, configuration, and account data are formatted into log messages and forwarded to the network logging facilities for recording on the network log file. Failure information, configuration changes, and some accounting data (see Accounting Database Restrictions later in this chapter) are collected on an event occurring basis. NPA uses the data recorded on the network log file to produce reports about the network and its operation.

Components called Log Management Entities (Log MEs) provide the logging function. The Log MEs ensure that information is recorded on a mass storage log file on the appropriate host.

Each DI contains a Dependent Log Management Entity (Dependent Log ME), which receives log messages from the data collection software in the DI. The Dependent Log ME then tests to see whether or not a log message should be written to the network log file. Network configuration commands determine which log messages should be written to the network log file. If the answer to the Dependent Log ME test is yes, a log message is routed to the Independent Log Management Entity (Independent Log ME). If the answer is no, the log message is discarded.

The Independent Log ME receives log messages from one or more Dependent Log MEs. The Independent Log ME is responsible for writing log messages to the network log file.

Each log message is assigned an identifying number and can be selectively logged. The log message identification defines the type of information the log message contains and determines to which log file(s) the log message is written (currently only one log file).

Data Reformatting

The organization of network log data is performed by the data reformatting utility REFORMAT_CDCNET_LOG_FILE (REFCLF) (figure 5-1). REFCLF accesses, reads, and organizes the data contained in the network log files on your local host. You must transfer network log files from other hosts (if you have more than one) in your network to the local host prior to using this utility. Use standard file transfer capabilities to accomplish file transfer.

NOTE

The transfer of files can only be done on like hosts (for example, NOS to NOS).

When your system is initialized, the NPA database files are empty permanent files. At startup, your system generates statistics (see Statistics Information later in this chapter) and data that are logged into the network log file on your local host.

REFCLF reads your network log files, finds and extracts the appropriate log data, and then reformats and writes the appropriate log data to the data files. Collectively, these files are known as the NPA database.

The NPA database contains information for subsequent use by the network reporting functions.

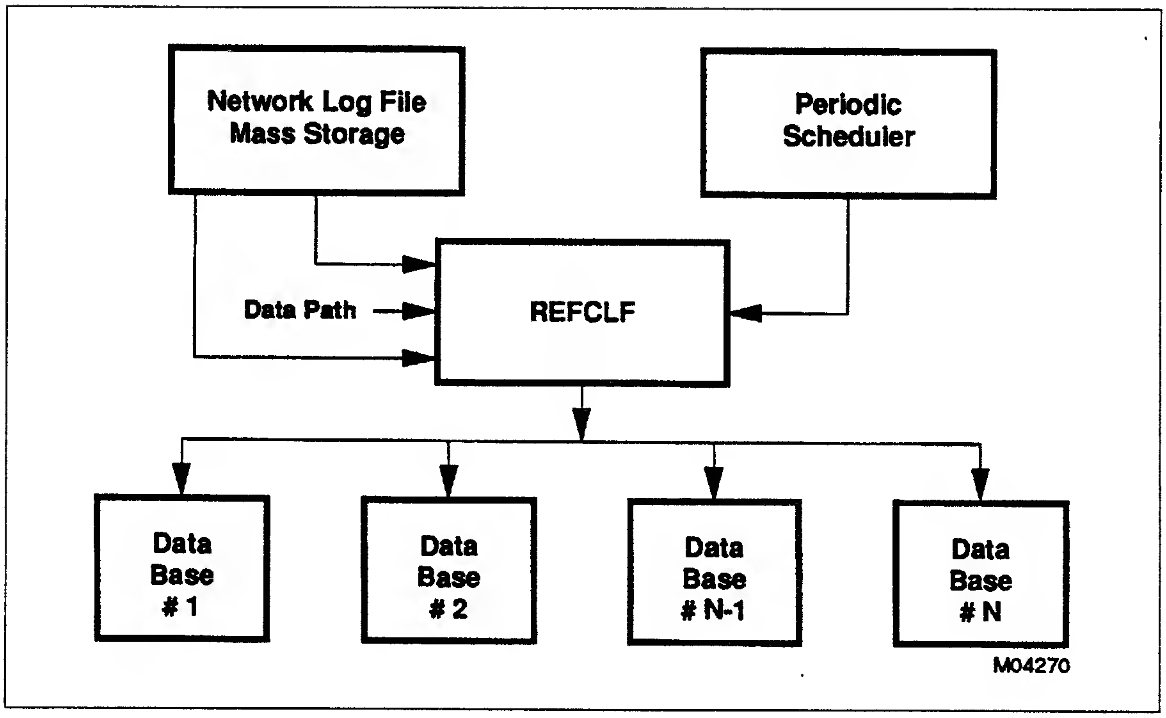


Figure 5-1. Data Reformatting Process

Data Reporting

The primary function of data reporting is to generate reports upon request. For the NPA report analysis tool, an existing commercial report generator is used so that more NPA development effort can be expended in the area of intelligent network analysis, rather than in duplicating yet another effort of report formatting.

Error prioritization is reported according to system impact consequence. Critical errors are reported first and informative errors last. The grouping of certain types of errors produces concise and relevant reports. This feature permits smaller sites with fixed or minimal maintenance resources to allocate them in a more cost-efficient manner.

The required capabilities of a flexible report writer combined with the desired data management facilities to sort data according to defined criteria are performed by the Information Processing Family Version 2 (IPF2). For more information on IPF2, see chapter 7 of this manual.

You use the NPA command CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) to generate reports.

Accounting Database Restrictions

NPA uniquely processes accounting data written as log messages to the network log file. REFCLF selects the accounting log messages to write to the accounting database NPBACNT, but does not reformat them in the same manner as other log messages. The accounting database basically reflects the raw accounting log messages. NPA does not provide additional services for processing accounting data (no reports are provided; accounting data cannot be archived or reloaded). The processing of the accounting database is left strictly up to the user.

NOTE

Detailed information on the file structure is available through SOLVER, an online facility for reporting problems.

Statistics Control

CDCNET statistics are numerical indicators of network performance. They include counts of data traffic and various events detected by the CDCNET communications software. Some examples of statistics include the number of messages or characters transmitted or received per line or DI and the number of errors encountered during a sampling period. You can use statistics to determine how the network is performing and to identify potential or real problems such as failing software processes or communication bottlenecks on lines and network solutions.

You may gather CDCNET statistics using statistics control commands. These commands start and stop the collection and reporting of statistics for the following network components: network solutions, communication lines, and software processes (such as the file access management function, log message recording, and gateways). There are start and stop commands for each of the three types of components for which you may gather statistics.

Statistics collection is started for the three types of statistics that may be collected (line, network, and process) by the start metrics commands. Once started, statistics are gathered over a collection period called a report interval. The report interval is set by a parameter on each start metrics command. This interval may differ between the components you are sampling. Collection of statistics is continuous; when one interval ends, another starts. At the end of a report interval, the statistics gathered during the report interval are reported by a log message, which is placed in the network log file, and a new report interval begins.

You may stop the collection and reporting of statistics by the stop metrics commands. These commands may be entered either before or after you obtain the statistics results (see Obtaining Statistics Results later in this chapter).

The appropriate log messages must be enabled for you to receive statistics information. The default set of log messages enabled by CDCNET includes the appropriate log messages providing statistical information.

Statistics Groups

There are three levels of statistics that are collected: summary, expanded, and debug statistics.

Summary statistics provide an overview of the operation of a line, network solution, or software process. Examples include the number of messages received and characters transmitted. In most cases, summary statistics provide sufficient statistical information about a component's performance.

Expanded statistics are a refinement of summary statistics. Examples include response times for a terminal user, number of messages processed for each user, and distribution of size of messages transmitted and received by a software component. Expanded statistics are useful in cases where a service is being provided for an individual user through a connection, because they can give more specific information about the connection and how the service is performing using a particular connection. In contrast, summary statistics provide an overview of how the service is working for all users. Not every component supports expanded statistics.

Debug statistics are a further refinement of statistics and include information that can be used to debug software components. Examples include the amount of global memory used and memory addresses involved. Not every statistic type has expanded and debug levels; only process statistics have the debug statistics group.

Example statistics groups are in the statistics gathered for an HDLC interface. HDLC interface statistics report character, frames, and message information daily or hourly.

Statistics levels are not hierarchical. You can start collection of expanded or debug statistics without also starting summary statistics collection. The default group level for all start metrics commands is summary statistics. The default for all stop metrics commands is to stop all statistics groups. When you stop statistics collection and reporting by specifying groups, any statistics groups not specified in the command remain in effect. However, if you send a start metrics command and have all statistics groups reporting, and later stop statistics without specifying all groups, any groups not specified continue to be collected and reported.

Starting and Stopping Statistics

1. Start the statistics using one or all of the following start metrics commands (send the commands within SEND_COMMAND).

    START_LINE_METRICS
    START_TRUNK_METRICS
    START_PROCESS_METRICS

    ```
    senc c='start_line_metrics line_name=line31..
    report_interval=300',s=west_tdi

    senc c='start_trunk_metrics network_name=ether1..
    report_interval=300',s=mdi1

    senc c='start_process_metrics process=osi_transport..
    report_interval=300',s=tdi_3
    ```

2. Enter one or all of the following stop metrics commands either before or after obtaining the statistics results.

    STOP_LINE_METRICS
    STOP_TRUNK_METRICS
    STOP_PROCESS_METRICS

    ```
    senc c='stop_line_metrics line_name=line31',s=west_tdi

    senc c='stop_trunk_metrics network_name=ether1',s=mdi1

    senc c='stop_process_metrics process=osi_transport',s=tdi_3
    ```

Obtaining Statistics Results

CDCNET statistics are reported by log messages, which are written to the CDCNET network log file on the host computer. You can display the statistics by defining the log messages as alarm messages, using the DEFINE_SOURCE_ALARM_MESSAGE command.

To obtain statistics, reformat the CDCNET log file containing the statistics messages by using the REFORMAT_CDCNET_LOG_FILE (REFCLF) command. REFCLF reorganizes the network log file (a chronological list of all log messages generated by the network's DIs), and builds files of various types of log messages called databases. NPA has standard database types and names. Each database contains a certain type of log message, such as log messages for a DI's CPU and memory use, or messages relating to terminal and connection performance. These databases are used to develop statistics reports.

Statistics reports are created from the NPA databases by using the CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) command.

Log file reformatting and report generation (by the NPA commands REFCLF and CRECAR) may be done by running a routine batch job when the network and operating system are being shut down or started up.

While statistics are being reported, you can monitor statistics messages at your operations station by enabling the statistics messages as alarms. Use the DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM) command to enable the messages as alarms. The message numbers to enable for line, network, and process metrics are shown in table 5-1.

Table 5-1. Statistics Commands and Message Numbers

Command	Message Number
START_LINE_METRICS	166
START_PROCESS_METRICS	94, 291, 299, 405, 424, 446, 547, 737, 889, 890, 1357, 1453, 746, 1435, 1628, 1629, 1648, 1693, 1820, 1821, 1873
START_TRUNK_METRICS	562, 665, 639

How To Initiate NPA Reports

Use the following procedures to initiate NPA reports. An example of NPA report generation follows these procedure.

NOTE

If you are using NOS/VE, you can use the Concurrent Maintenance Library for the Virtual Environment (CML/VE) to interactively generate some of the NPA reports. See the CML/VE Reference manual or the CDCNET Hardware Installation and Troubleshooting manual for information on CML/VE usage.

If you are using NOS, you can use the Common Maintenance Software Interface (CMSI) to interactively generate some of the NPA reports. See the CML Reference manual or the CDCNET Hardware Installation and Troubleshooting manual for information on CMSI usage.

If you want to generate customized NPA reports, see chapter 7, How To Create Customized NPA Reports Using IPF2 Database Files.

NOS/VE Only

1. Log in to the host computer by entering your required user name and password. If you successfully log in, your terminal displays a slash.

   ```
   /
   ```

2. You make NPA available to your job with the following SCL command.

   ```
   CREATE_COMMAND_LIST_ENTRY $SYSTEM.CDCNET.VERSION_INDEPENDENT.COMMAND_LIBRARY
   ```

3. You then enter the NPA command. This command does all of the setup necessary to run NPA. The default attribute file NPAATTR is made local with this command along with the necessary library files needed for NPA command execution.

   ```
   npa
   ```

4. When the following prompt appears, you are ready to enter NPA commands.

   ```
   np/
   ```

5. Execute the REFORMAT_CDCNET_LOG_FILE (REFCLF) command. REFCLF is used to execute the reformatting process. REFCLF converts network log file records into database records and transfers the converted records into appropriate data files.

   In this example, REFCLF processes up to 30 cycles of the $SYSTEM.CDCNET.LOG file.

   ```
   REFCLF,BD=850903,BT=120000,ED=,..
   850907,DB=(ACNT,CONN,ETHR,SERR),O=REFREP,LFL=PURGLST
   ```

   NOTE

   In order to execute the REFCLF command, you must have access to the network log file(s). If you do not have access to this file, have the network administrator execute the REFCLF command.

6. Execute the CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) command. CRECAR is used to generate NPA reports.

```
CRECAR,RN=ETHRRP2,BD=850101,ED=850102
```

7. The generated report is stored in file CREOUT. To view this report at your terminal, enter:

```
EDIF,CREOUT
```

8. To exit the utility (this disables NPA commands, and the np/ prompt no longer appears), enter:

```
quit
```

NOS Only

1. Log in to the host computer by entering your required user name and password. If you successfully log in, your terminal displays a slash.

```
/
```

2. You then enter NPA. This command does all of the setup necessary to run NPA. The default attribute file NPAATTR is made local with this command along with the necessary library files needed for NPA command execution.

```
npa
```

3. When a slash appears, you are ready to enter NPA commands.

```
/
```

4. Execute the REFORMAT_CDCNET_LOG_FILE (REFCLF) command. REFCLF is used to execute the reformatting process. REFCLF converts network log file records into database records and transfers the converted records into appropriate data files.

```
REFCLF.LF=(NLA0225,NLA0226),BD=850903,BT=120000,ED=,..
850907,DB=(ACNT,CONN,ETHR,SERR),O=REFREP,LFL=PURGLST
```

NOTE

In order to execute the REFCLF command, you must have access to the network log file(s). If you do not have access to this file, have the network administrator execute the REFCLF command.

REFCLF does not support full screen execution.

5. Execute the CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) command. CRECAR is used to generate NPA reports.

```
CRECAR,RN=ETHRRP2,BD=850101,ED=850102,APPEND=yes
```

6. The generated report is stored in file CREOUT. To view this report at your terminal, enter:

```
FSE,CREOUT
```

NPA Report Generation Example

This NOS example creates a hardware status report (HRDWRP1) similar to the one shown in figures 5-2 and 5-3. Figure 5-2 shows the report heading page. Figure 5-3 shows the actual report data page.

To create the report:

1. Log in to your terminal and access NPA by entering:

   ```
   NPA
   ```

2. Convert the network log file records into database records by running the REFCLF command. You can only perform this step if you have access to the private network log file.

   In this example, REFCLF processes up to 30 files beginning with NL or NETLF.

   ```
   REFCLF.DB=HERR,DBFUN=NETADMN
   ```

3. Create the NPA hardware status report (HRDWRP1) using the line mode command entry procedure for CRECAR by entering:

   ```
   CRECAR,RN=HRDWRP1,DBFUN=NETADMN
   ```

4. The hardware report HRDWRP1 is created and stored in file CREOUT. To view the report (figure 5-2 and figure 5-3) on your terminal, enter:

   ```
   FSE,CREOUT
   ```

```
                                                        87/09/15

                   NETWORK PERFORMANCE ANALYZER
                       VERSION 1.10/3403

                     CDCNET HARDWARE MESSAGES
                     SORTED BY DATE AND TIME

                          HRDWRP1 REPORT

           TIME PERIOD = 00/01/01 0000 - 99/12/31 2400

                     SYSTEM ID SELECTED = ALL

                     LOG IDS SELECTED   = ALL

                     LOG IDS EXCLUDED   = ALL

                     SEVERITY SELECTED  = CATASTROPHIC
                                          FATAL
                                          ERROR
                                          WARNING
                                          INFORMATIVE
```

Figure 5-2. HRDWRP1 Report Heading Page

```
REPORT DAY: 86/01/01                                                            PAGE      1
                                      HROWRP1 REPORT
                                  START TIME = 0000 HOURS

  DATE       TIME       SYSTEM IO     LOG IO SEVERITY
======== ========== ============  ====== ============

86/01/01 00.00.00927 0800253000A2      338      ERROR
--ERROR--  MPB FAILED ON-BOARO TESTING
BEFORE INITIALIZATION WAS SUCCESSFUL.
SLOT NUMBER= 0
FATAL ERRORS= 7

86/01/01 00.00.00930 0800253000A2      340      ERROR
--ERROR--  PMM HAO RECOVEREO PARITY ERRORS
DURING ON-BOARD TESTING.
SLOT NUMBER= 1
ERRORS= 39168
FIRST FAILING AODRESS= 00010000

86/01/01 00.00.00933 0800253000A2      341      ERROR
--ERROR--  SMM SINGLE BIT ERRORS OCCURREO
OURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 1942
ERROR LOG= 0648

86/01/01 00.00.00935 0800253000A2      342      ERROR
--ERROR--  SMM MULTIPLE BIT ERRORS OCCURREO
OURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 11671
ERROR LOG= 0473

86/01/01 00.00.55028 0800253000A2       19   INFORMATIVE
CONFIGURATION COMPLETE,
CONFIGURATION FILE SOURCE:
NETWORK ID: 41454646, SYSTEM IO: 080025300OBE
```

Figure 5-3. HRDWRP1 Report Data Page

How To Enter NPA Commands in Screen Mode Format (NOS Only)

If you are using NOS and your terminal is operating in screen mode, the system can use full screen displays to prompt you for parameters and provide you with information to help you execute NPA commands.

NOTE

All NOS NPA commands can use full screen displays except REFCLF which operates in line mode format only.

ARCNDB

Use ARCNDB to remove records from the NPA databases and archive this information into an archive file. To enter this command interactively using screen mode, do the following steps.

1. Enter:

```
ARCNDB
```

2. Press the RETURN key, and the following screen appears:

```
            ARCHIVE NPA DATA BASES

ARCHIVE FILE              :______
ARCHIVE FILE USER NAME:______
DATA BASE                .:______
DBFUN                     :______
BEGIN DATE (YYMMDD)       :______
BEGIN TIME (HHMMSS)       :______
END DATE   (YYMMDD)       :______
END TIME   (HHMMSS)       :______
OUTPUT                    :______
BEGINNING DATE OFFSET :______
ENDING DATE OFFSET        :______

             Specify values and press NEXT when ready
```

3. A cursor appears on the ARCHIVE FILE line to prompt you to enter this parameter first. The file name is a required parameter and must be entered. If you fail to enter the ARCHIVE FILE parameter, the following prompt appears:

```
Please enter ARCHIVE FILE
```

   If you enter an illegal ARCHIVE FILE, the following prompt appears:

```
ILLEGAL FILE NAME ** illegal name **
  ARCHIVE PROCESSING ABORTED
```

4. Enter parameter values. Press the TAB key to move the cursor to the next parameter line. Use the arrow keys to position the cursor at any parameter line on which you wish to make an entry.

5. Press the RETURN key after you have entered all of your desired parameters, and the selected database(s) are archived into the archive file.

Example:

This example archives all the existing database files into the archive file NPAARC. The remaining parameters, which are all optional, are assigned default values.

Begin by entering the archive file name:

```
ARCHIVE FILE: NPAARC
```

Use the arrow key to position the cursor on the database line. Enter the database:

```
DATA BASE: ALL
```

Press the RETURN key, and all databases are archived into the archive file NPAARC.

CHAEOL

Use CHAEOL to replace existing expected operating limits. To enter this command interactively in screen mode:

1. Enter:

   ```
   CHAEOL
   ```

2. Press the RETURN key, and the following screen appears:

   ```
    CHANGE EXPECTED OPERATING LIMITS PROCEDURE

   REPORT NAME TO CHANGE:_______________

      Specify values and press NEXT when ready
   ```

3. A cursor appears on the REPORT NAME TO CHANGE line to prompt you to enter this parameter. Enter your desired report name from the list of valid keywords that appears in the CHAEOL procedure. If you insert an illegal REPORT NAME, the following prompt appears:

   ```
   Please correct (illegal name)
   ```

4. A cursor appears on the first line that identifies an NPA expected operating limit. Change the current limit shown on any line by entering a new number over the existing number. Move the cursor from the first line to the next by pressing the TAB key, or position the cursor to any line you desire by using the arrow keys.

Example:

You wish to change the NPA expected operating limits for report ETHRRP2.

Enter ETHRRP2 on the REPORT NAME TO CHANGE line.

```
 CHANGE EXPECTED OPERATING LIMITS PROCEDURE

REPORT NAME TO CHANGE:ETHRRP2________
```

Press the RETURN key, and the following screen appears:

```
                    ETHERNET REPORT 2 EOL UPDATE PROCEDURE

                                        LOW1 LOWER CRC ERROR LIMIT (PER HOUR): 01
                                       HIGH1 UPPER CRC ERROR LIMIT (PER HOUR): 04
                                  LOW2 LOWER ALIGNMENT ERROR LIMIT (PER HOUR): 01
                                 HIGH2 UPPER ALIGNMENT ERROR LIMIT (PER HOUR): 05
                                    LOW3 LOWER OVERRUN ERROR LIMIT (PER HOUR): 01
                                   HIGH3 UPPER OVERRUN ERROR LIMIT (PER HOUR): 07
                                   LOW4 LOWER RESOURCE ERROR LIMIT (PER HOUR): 01
                                  HIGH4 UPPER RESOURCE ERROR LIMIT (PER HOUR): 08
 LOW5 LOWER ABNORMAL COLLISION DETECTION LOGIC ERROR LIMIT (PER HOUR): 01
HIGH5 UPPER ABNORMAL COLLISION DETECTION LOGIC ERROR LIMIT (PER HOUR): 09

                    Specify values and press NEXT when ready
-----------------------------------EOLET2---------------------------------

THIS PROCEDURE WILL CHANGE THE EOL FOR THE ETHERNET ERROR STATISTICS.
```

You wish to change the current values for HIGH1 UPPER CRC ERROR LIMIT, HIGH3 UPPER OVERRUN ERROR LIMIT, and HIGH5 UPPER ABNORMAL COLLISION DETECTION LOGIC ERROR LIMIT on report ETHRRP2 to 7, 9, and 11, respectively.

The cursor appears on the first line. You do not want to change the limit shown on the first line. Therefore, press the TAB key or use the arrow keys to position the cursor on the second line, HIGH1 UPPER CRC ERROR LIMIT.

Change the current value of 04 to the new value of 7 by typing 07 over the current value.

Each time the TAB function executes, the cursor drops one line. Therefore, press the TAB key four times to move the cursor down to the HIGH3 UPPER OVERRUN ERROR LIMIT line. Or, use the arrow keys to position the cursor on the HIGH3 UPPER OVERRUN ERROR LIMIT line.

Enter your new value of 09 over the current value of 07. Press the TAB key four times to move down to the HIGH5 UPPER ABNORMAL COLLISION DETECTION LOGIC ERROR LIMIT line. Or, use the arrow keys to position the cursor on the HIGH5 UPPER ABNORMAL COLLISION DETECTION LOGIC ERROR LIMIT line. Enter the new value of 11 over the current value of 09.

Press the RETURN key, and CHAEOL replaces the old limits with the new ones you have entered.

CRECAR

Use CRECAR to generate reports. To enter this command interactively in screen mode, do the following steps.

1. Enter:

```
CRECAR
```

2. Press the RETURN key and the following screen appears:

```
 CREATE CDCNET ANALYSIS REPORT

REPORT NAME               :_______________
DATA BASE FILE USER NAME:_______________
BEGINNING DATE (YYMMDD)  :_______________
BEGINNING TIME (HHMM)    :_______________
ENDING DATE    (YYMMDD)  :_______________
ENDING TIME    (HHMM)    :_______________
BEGINNING DATE OFFSET    :_______________
ENDING DATE OFFSET       :_______________
SYSTEM ID      (XXXXXX)  :_______________
NAME OF OUTPUT FILE      :_______________
APPEND REPORT TO OUTPUT?:_______________
LOG ID         (XXXXX)   :_______________
EXCLUDE LOG ID (XXXXX)   :_______________
SEVERITY (I,W,E,F,C)     :_______________
COMPRESS REPORT?         :_______________

      Specify values and press NEXT when ready
```

3. A cursor appears on the REPORT NAME line to prompt you to enter this parameter first. Enter your desired report name from the list of valid keywords that appears in the CRECAR procedure. If you fail to enter a REPORT NAME, the following prompt appears:

```
Please enter REPORT NAME
```

If you enter an illegal REPORT NAME, the following prompt appears:

```
Please correct (illegal name)
```

4. REPORT NAME is the only required parameter for the CRECAR procedure. You may generate your chosen report, or report set, by pressing the RETURN key. If you do not enter any of the optional parameters, the report or reports generated are based upon the default parameter values defined in the CRECAR procedure description.

   To enter any or all of the optional parameters, use the TAB key or the arrow (cursor control) keys on your terminal to move the cursor to the desired parameter lines. When you press the TAB key, the cursor moves to the next parameter line. When you use the arrow keys, you can position the cursor at any parameter line on which you wish to make an entry.

5. Press the RETURN key after you have entered all of your desired parameters. CRECAR generates the report or reports you have requested based on the parameter values you have entered.

   Example:

   You wish to generate an HRDWRP1 report to receive a hardware message from all network device interfaces (DIs).

   Begin by entering the report name on the REPORT NAME line:

```
 CREATE CDCNET ANALYSIS REPORT

REPORT NAME        :HRDWRP1
```

If you wish to enter any of the other parameters instead of using the default value, press the TAB key until the cursor appears on the line of the parameter you wish to enter.

When you are ready to generate the report, press the RETURN key.

EDICLM

Use EDICLM to edit the explanatory information about CDCNET log messages (provided by NPA procedure EXPCLM). To enter this command interactively in screen mode:

1. Enter:

```
EDICLM
```

2. Press the RETURN key and the following screen appears:

```
                 EDIT CDCNET LOG MESSAGE

MESSAGE NUMBER (XXXXX)    :__________
EDITOR (FSE,EDIT OR XEDIT):__________

          Specify values and press NEXT when ready
```

3. The cursor appears on the MESSAGE NUMBER line to prompt you to enter the number of the log message that identifies the log message explanation you are going to edit.

   Enter the log message number and press the TAB key, and the cursor moves to the EDITOR line. Or, use the arrow keys to position the cursor on the EDITOR line.

   Enter the editing mode you want to use: Full Screen Editor (FSE), EDIT, or XEDIT. Press the RETURN key, and the log message explanation appears on your screen.

   If you plan to use FSE, you need not enter a parameter on the EDITOR line. Enter only the message number, and press the RETURN key to create the log message explanation to your screen.

   NOTE

   Only the Site Information portion of the log message explanation is to be edited.

Example:

You want to edit the Site Information portion of log message explanation 00001 using FSE.

Enter 00001 on the MESSAGE NUMBER line and press the RETURN key. (If you are using FSE, you do not need to make an entry on the EDITOR line.)

The following screen appears:

```
NOS FULL SCREEN EDITOR
Upper Case File NPATMP1  Lines 1 Thru 23 Size 23
.PROC,E00001*I"MESSAGE NUMBER 00001 EXPLANATION".
.HELP.

          LOG MESSAGE NAME

    00001  - - - ->      S_A_LOCAL_RECOVERY_FAILURE

        L O G  M E S S A G E  P U R P O S E

             THIS LOG MESSAGE INDICATES THAT A FAILURE OCCURRED WHILE EXECUTING
             A FAILED TASK'S RECOVERY PROCEDURE.

        A C T I O N  R E Q U I R E D

             A CDCNET ANALYST SHOULD BE NOTIFIED WITH THE INFORMATION REGARDING
             THE FAILED PROGRAM MODULE TO DETERMINE THE CONDITION OF THE
             CURRENT SYSTEM.

        S I T E  I N F O R M A T I O N

        .ENDHELP
        $REVERT,NOLIST.
```

Edit the Site Information portion of the message explanation and execute the QUIT function. The information (your additions, deletions, or changes) is permanent.

NOTE

In order to view the entire display, you might have to press the F3 key to see the next screen.

EXPCLM

Use EXPCLM to generate explanatory information about CDCNET log messages. To enter this command interactively in screen mode:

1. Enter:

```
EXPCLM
```

2. Press the RETURN key, and the following screen appears:

```
          EXPLAIN CDCNET LOG MESSAGE

MESSAGE NUMBER (XXXXX) :__________

     Specify values and press NEXT when ready
```

3. The cursor appears on the MESSAGE NUMBER line to prompt you to enter the number of the log message for which you want an explanation. Enter the log message number and press the RETURN key, and a full screen explanation appears. If you enter an illegal MESSAGE NUMBER, the following prompt appears:

```
Please correct (illegal number)
```

Example:

You want an expanded explanation of log message number 00001. Enter 00001 on the MESSAGE NUMBER line.

```
MESSAGE NUMBER (XXXXX) :00001
```

Press the RETURN key, and the following screen appears:

```
                              MESSAGE NUMBER 00001 EXPLANATION

                           Specify values and press NEXT when ready

-----------------------------------E00001---------------------------------

        LOG MESSAGE NAME

    00001  - - - ->       S_A_LOCAL_RECOVERY_FAILURE

       L O G   M E S S A G E   P U R P O S E

            THIS LOG MESSAGE INDICATES THAT A FAILURE OCCURRED WHILE EXECUTING
            A FAILED TASK'S RECOVERY PROCEDURE.

       A C T I O N   R E Q U I R E D

            A CDCNET ANALYST SHOULD BE NOTIFIED WITH THE INFORMATION REGARDING
            THE FAILED PROGRAM MODULE TO DETERMINE THE CONDITION OF THE
            CURRENT SYSTEM.

       S I T E   I N F O R M A T I O N
```

NOTE

In order to view the entire display, you might have to press the F3 key to see the next screen.

RELNDB

Use RELNDB to reload records from an archive file into the NPA databases. To enter this command interactively using screen mode, do the following steps.

1. Enter:

```
RELNDB
```

2. Press the RETURN key, and the following screen appears:

```
          RELOAD NPA DATA BASES

ARCHIVE FILE                :______
ARCHIVE FILE USER NAME:______
DATA BASE                   :______
DBFUN                       :______
BEGIN DATE (YYMMDD)         :______
BEGIN TIME (HHMMSS)         :______
END DATE   (YYMMDD)         :______
END TIME   (HHMMSS)         :______
OUTPUT                      :______

           Specify values and press NEXT when ready
```

3. A cursor appears on the ARCHIVE FILE line to prompt you to enter this parameter first. The file name is a required parameter and must be entered. If you fail to enter the ARCHIVE FILE, the following prompt appears:

   ```
   Please enter ARCHIVE FILE
   ```

   If you enter an illegal ARCHIVE FILE, the following prompt appears:

   ```
   ILLEGAL FILE NAME  ** illegal name **
   RELOAD PROCESSING ABORTED
   ```

   Use the TAB key or the arrow keys on your terminal to move the cursor to the desired parameter line. When you press the TAB key, the cursor moves to the next parameter line. When you use the arrow keys, you can position the cursor at any parameter line on which you wish to make an entry.

   Press the RETURN key after you have entered all of your desired parameters. The selected database(s) are reloaded into the CDCNET log file.

Example:

This example reloads all the existing database files from the archive file NPAARC into the CDCNET log file. The remaining parameters, which are all optional, are assigned default values.

Begin by entering the archive file name:

```
ARCHIVE FILE: NPAARC
```

Use the arrow key and position the cursor on the database line. Enter the database:

```
DATA BASE: ALL
```

Press the RETURN key, and the archive file is reloaded into the database.

How to Enter NPA Commands in Line Mode Format

If your terminal is in line or screen mode, you can execute NPA commands by entering the desired command name, all of the required parameters, and any desired optional parameters for that command, followed by a carriage return. Default values are assigned to the omitted optional parameters.

If you omit a required parameter when using NOS, the terminal reverts back to the Screen Mode Parameter List display (except for REFCLF which uses line mode format only). You must then select the required parameters as you would in screen mode. When this occurs, see How to Enter NPA Commands in Screen Mode Format for parameter entry procedures.

If you omit a required parameter when using NOS/VE, a prompt message appears indicating which parameter is required. The following examples show how to enter the NPA commands in line mode.

ARCNDB

Use ARCNDB to remove records from the NPA databases and archive this information into an archive file.

1. To enter this command in line mode format, enter the following:

   ```
   ARCNDB,AF=filename,DB=value
   ```

   The ARCHIVE_FILE parameter is required and must be entered. If you omit the required parameter and you are using NOS, the screen displays the screen mode parameter entry display (see the ARCNDB screen mode procedure to enter from this display). If you omit the required parameter and you are using NOS/VE, a prompt message appears indicating which required parameter has been omitted. The remaining optional parameters are assigned default values if you do not enter them.

2. After entering the parameters, press the RETURN key and the selected database(s) are archived.

Example:

This example archives all the existing database files into the archive file NPAARC. All optional parameters are assigned default values.

```
ARCNDB,AF=NPAARC,DB=ALL
```

Press the RETURN key to execute.

CHAEOL

NOTE

When CHAEOL is entered in line mode format, the terminal must be in screen mode to enable the limits display and limits value updating.

Use CHAEOL to replace existing expected operating limits. For example, to change the limits on the ETHRRP2 report, perform the following steps.

1. Enter the CHAEOL command and the REPORT_ NAME parameter.

   ```
   CHAEOL,RN=ethrrp2
   ```

   The REPORT_NAME (RN) parameter is required and must be entered. If you omit the required REPORT_NAME (RN) parameter and you are using NOS, the screen displays the screen mode parameter entry display. When you omit a required parameter using NOS/VE, a prompt message appears indicating which parameter is required. The remaining optional parameters are assigned default values if you do not enter them.

2. Press the RETURN key and the operating limits for the selected report appear as in the following NOS/VE example:

   ```
    20  "maximum_crc_error            "
     0  "minimum_alignment_error      "
    30  "maximum_alignment_error      "
     0  "minimum_overrun_error        "
    40  "maximum_overrun_error        "
     0  "minimum_resource_error       "
   100  "maximum_resource_error       "
     0  "minimum_abnormal_logic       "
     3  "maximum_abnormal_logic       "
   ```

3. A cursor appears on the first line that identifies an NPA expected operating limit. Move the cursor from the first line to the next by pressing the TAB key, or position the cursor to any line you desire by using the arrow keys. Change the current limit shown on any line by entering a new number over the existing number.

CRECAR

Use the NPA report generator CRECAR to generate reports.

1. To enter this command in line mode format, key in the following:

   ```
   CRECAR,RN=report name
   ```

   The REPORT_NAME (RN) parameter is required and must be entered. If you omit the required REPORT_NAME (RN) parameter and you are using NOS, the screen displays the screen mode parameter entry display (see the CRECAR screen mode procedure to enter from this display). When you omit a required parameter using NOS/VE, a prompt message appears indicating which parameter is required. The remaining optional parameters are assigned default values if you do not enter them.

2. After entering the parameters, press the RETURN key and the report is generated.

Example:

This line mode example creates the HRDWRP1 report with all optional parameters set to default values:

```
CRECAR,RN=HRDWRP1
```

Press the RETURN key to execute, and the HRDWRP1 report is placed in file CREOUT.

EDICLM

Use the NPA help utility procedure EDICLM to edit the explanatory information (Site Information only) about CDCNET log messages provided by procedure EXPCLM.

1. To enter this command in line mode format, key in the following:

   ```
   EDICLM,MN=log id number
   ```

   The MESSAGE_NUMBER (MN) parameter is required and must be entered. If you omit the required parameter and you are using NOS, the screen displays the screen mode parameter entry display (see the EDICLM screen mode procedure to enter from this display). If you omit the required parameter and you are using NOS/VE, a prompt message appears indicating which parameter is required. The remaining optional parameter is assigned the default value if you do not enter one.

2. After entering the parameters, press the RETURN key to execute.

Example:

In this NOS example, log message number 00001 appears on your terminal screen and you may edit the Site Information portion of the message using FSE (default value).

```
EDICLM,MN=00001
```

Press the RETURN key, and the following message appears (you may have to press F3 to view the entire screen):

```
NOS FULL SCREEN EDITOR
Upper Case File NPATMP1  Lines 1 Thru 23 Size 23
.PROC,E00001*I"MESSAGE NUMBER 00001 EXPLANATION".
.HELP.

          LOG MESSAGE NAME

     00001  - - - ->      S_A_LOCAL_RECOVERY_FAILURE

        L O G   M E S S A G E   P U R P O S E

             THIS LOG MESSAGE INDICATES THAT A FAILURE OCCURRED WHILE
             EXECUTING A FAILED TASK'S RECOVERY PROCEDURE.

        A C T I O N   R E Q U I R E D

             A CDCNET ANALYST SHOULD BE NOTIFIED WITH THE INFORMATION
             REGARDING THE FAILED PROGRAM MODULE TO DETERMINE THE CONDITION
             OF THE CURRENT SYSTEM.

        S I T E   I N F O R M A T I O N

        .ENDHELP
        $REVERT,NOLIST.
```

Edit the Site Information portion of the message explanation and execute the QUIT function. The information (your additions, deletions, or changes) is permanent.

EXPCLM

Use the NPA help utility procedure EXPCLM to generate explanatory information about CDCNET log messages.

1. To enter this command in line mode format, key in the following:

```
EXPCLM,MN=log id number
```

   The MESSAGE_NUMBER (MN) parameter is required and must be entered. If you omit the required parameter and you are using NOS, the screen displays the screen mode parameter entry display (see the EXPCLM screen mode procedure to enter from this display). If you omit a required parameter and you are using NOS/VE, a prompt message appears indicating which required parameter has been omitted. The remaining optional parameter is assigned the default value if you do not enter one.

2. After entering the parameter, press the RETURN key to execute.

Example:

In this NOS example, log message 00001 appears on your terminal screen:

```
EXPCLM,MN=00001
```

Press the RETURN key to execute, and the following message appears:

```
------------------------------------E00001----------------------------------

        LOG MESSAGE NAME

    00001  - - - ->       S_A_LOCAL_RECOVERY_FAILURE

       L O G   M E S S A G E   P U R P O S E

            THIS LOG MESSAGE INDICATES THAT A FAILURE OCCURRED WHILE EXECUTING
            A FAILED TASK'S RECOVERY PROCEDURE.

       A C T I O N   R E Q U I R E D

            A CDCNET ANALYST SHOULD BE NOTIFIED WITH THE INFORMATION REGARDING
            THE FAILED PROGRAM MODULE TO DETERMINE THE CONDITION OF THE
            CURRENT SYSTEM.

       S I T E   I N F O R M A T I O N
```

NOTE

In order to view the entire display, you might have to press the F3 key to see the next screen.

REFCLF

Use the NPA reformatting process REFCLF to convert network log file records into database records.

1. To enter this command, key in the following:

   ```
   REFCLF DB=value (NOS/VE)
   ```

   or

   ```
   REFCLF.DB=value (NOS)
   ```

   Either the DATA_BASE (DB) or USER (U) parameter must be entered or REFCLF terminates with an error message. The remaining optional parameters are assigned default values if not entered.

   NOTE

   In order to use REFCLF, you must have access to the network log files.

2. Press the RETURN key and the network log file records are reformatted into database records.

RELNDB

Use RELNDB to reload records from an archive file into existing NPA databases.

1. To enter this command in line mode, enter the following:

   ```
   RELNDB,AF=filename,DB=value
   ```

   The ARCHIVE_FILE (AF) parameter is required and must be entered. If you omit this parameter and you are running NOS, the screen displays the screen mode parameter entry display (see the RELNDB screen mode procedure to enter from this display). If you omit the required parameter and you are using NOS/VE, a prompt message appears indicating which required parameter has been omitted. The remaining optional parameters are assigned default values if you do not enter them.

2. After entering the parameters, press the RETURN key to execute.

Example:

This example reloads all the existing database files in the archive file NPAARC to the log file. All optional parameters are assigned default values.

```
RELNDB,AF=NPAARC,DB=ALL
```

Press the RETURN key to execute.

How To Get Help on NPA Procedures

After you call a procedure interactively, but before the system executes the specified procedure, you can have a dialogue with the system about the procedure.

NOTE

These procedures are for getting help on commands and parameters. If you want help on log messages, see the EXPCLM command in this chapter.

How To Get Help in Screen Mode (NOS Only)

In screen mode, you press the HELP key (F5 key on some terminals) to initiate a dialogue with the system. Using this key allows you to:

- Reenter parameter values that are in error
- Request information about a procedure parameter
- Request information about the procedure itself

For example, if you are using the CRECAR procedure in screen mode and you enter an illegal value for the REPORT_NAME parameter, the system informs you of your error by displaying the following message at the top of your screen:

```
PLEASE CORRECT (erroneous value)
```

To find out why your entry was illegal, ask the system by pressing the HELP key.

A list of legal report names appears at the bottom of your screen. Now you may enter a legal value for the REPORT_NAME parameter.

If you again press the HELP key, you receive the following information about the CRECAR procedure at the bottom of your screen:

```
----------------------------CRECAR-------------------------------------------

THIS PROCEDURE WILL GENERATE NETWORK PERFORMANCE REPORTS.
```

How To Get Help in Line Mode (NOS Only)

If you enter an incorrect value for a parameter in line mode, the system informs you of your error by displaying a message.

For example, if you are using CRECAR and enter an illegal value for REPORT_NAME, the following message appears:

```
CREATE CDCNET ANALYSIS REPORT
CORRECT REPORT NAME  ?
```

Enter a report name or, for help, type in a question mark (?) and a list of legal report names is displayed followed by a second prompt for REPORT_NAME.

Retry entering a report name.

How To Get Help in Line Mode (NOS/VE Only)

To get help on any command and/or its parameters while using NPA on NOS/VE, enter the following command:

```
display_command_information command name.
```

or

```
disci command name.
```

For example, if you wanted help on the CRECAR command, enter the following:

```
disci crecar
```

The screen then displays the CRECAR command description and its parameters.

NPA Reports and Report Formats 6

NPA Reports and Report Formats 6

Common Report Format Features

All report formats assume an 80-column wide data window to permit 10 character-per-inch printers to produce standard sized, 8-1/2-inch wide by 11-inch long reports or line printer output on computer paper that can be cut down to an 8-1/2-inch width.

Each report consists of two parts; a one-page report heading, and a set of one or more report data pages.

NOTE

For more detailed information on the NPA commands used to generate these reports, see the CDCNET Commands Reference manual.

Report Headings

The first page of all uncompressed NPA reports is a report heading similar to that shown in figure 6-1. The date the report was generated is shown in the upper right corner. Centered on the page is the static information for the report. It includes the product name (NETWORK PERFORMANCE ANALYZER); version number of the NPA report module; report full title; report organization information; report brief title; and the date, time period, and system ID requested by the CRECAR parameters.

NOTE

The heading pages of reports can be eliminated by selecting the COMPRESS parameter of the CRECAR command.

Some reports also include the log IDs and/or the severity level. Figure 6-2 shows a report heading with the log IDs included. Figure 6-3 shows a report heading with the log IDs and the severity levels included.

NOTE

The example report headings were generated on NOS/VE. On NOS, the report heading pages show slightly different spacing between lines.

```
                                                              88/06/01




                          NETWORK PERFORMANCE ANALYZER
                              VERSION 1.10/5501

                            CDCNET ETHERNET STATISTICS
                        SORTED BY SID, DATE, AND CARD SLOT

                                 ETHRRP1 REPORT

                  TIME PERIOD = 00/01/01 0000 - 99/12/31 2400


                            SYSTEM IDS SELECTED = 080025100083
                                                  080025100078
```

Figure 6-1. Standard Report Heading

```
                                                        88/06/01




                    NETWORK PERFORMANCE ANALYZER
                         VERSION 1.10/5501

                    CDCNET CONFIGURATION MESSAGES
                    SORTED BY DI, DATE, AND TIME

                          CONFRP1 REPORT

            TIME PERIOD = 00/01/01 0000 - 99/12/31 2400


                     SYSTEM IDS SELECTED = 0800253000D2
                                           080025100083



                     LOG IDS SELECTED    = ALL


                     LOG IDS EXCLUDED    = NONE
```

Figure 6-2. Report Heading with Log IDs

```
                                                          88/06/01




                  NETWORK PERFORMANCE ANALYZER
                      VERSION 1.10/5501

                   CDCNET EVENT LOG MESSAGES
                    SORTED BY DATE AND TIME

                        EVNTRP1 REPORT

            TIME PERIOD = 00/01/01 0000 - 99/12/31 2400

                    SYSTEM IDS SELECTED = 0800253000D2
                                          080025100083
                                          0800253000A2


                    LOG IDS SELECTED    = IDS NOT EXCLUDED


                    LOG IDS EXCLUDED    = 00073


                    SEVERITY SELECTED   = CATASTROPHIC
                                          FATAL
                                          ERROR
                                          WARNING
                                          INFORMATIVE
```

Figure 6-3. Report Heading with Log IDs and Severity

Report Data Pages

An abbreviated page heading tops each report data page and contains report dynamic information, some of which is taken from the report data itself. The report day is shown in the left corner. The page number is shown in the right corner. Centered in the page heading is the report brief title. Underneath the brief title might be either variable report-dependent title information or, in some cases, no further information. On many reports, DI TITLE and SID are displayed beneath the report title.

NOTE

NPA reports on NOS are Control Data display code files. They may appear to contain some garbled data if not viewed interactively in NORMAL mode or if not printed appropriately.

Following the page heading, the content of NPA reports can follow one of two general formats. The first format consists of columns of data aligned under descriptive column headers. These reports usually contain DI statistical or summary data. This type of report may include predefined Control Data Expected Operating Limits (CDC EOL) as part of the column header. The limits are listed directly under the descriptive headers as two numbers. The upper CDC EOL is listed first with the lower CDC EOL listed under it on the next line. In cases where there is no CDC EOL for a particular column, no space is reserved for them. The following is an example showing the first general report format, including CDC EOLs.

```
REPORT DAY: 86/09/05                                                      PAGE     1
                                          ETHRRP2 REPORT
         TITLE = MTI_83                                       SID = 080025100083

                                          CRC       ALIGN.     OVERRUN   RESOURCE   ABNORMAL
                                        ERRORS     ERRORS      ERRORS     ERRORS     LOGIC
ENDING       FRAMES     FRAMES     [      20] [      30] [      40] [     100] [       3]
 TIME   CS RECEIVED      SENT      [       0] [       0] [       0] [       0] [       0]
====== == =========  =========  ========= ========= ========= ========= =========

 0924   6     13144      13581          0          0          0          0          0
 1024   6     13291      13704          0          4          0          0          0
 1224   6     11549      11782          0          0          0          0          0
 1424   6      8276       8175          0          0          0          0          0
 1524   6     13735      14154          0          2          0          0          0
```

The second general report format consists mainly of text lines following the page heading. A brief column header describes the fields within a text header that precedes the text body of the each message. This type of report is generated by NPA by combining DI logged data with static text contained in DI message templates. An example of this report format follows.

NOTE

Reports can be optionally compressed. Compressed reports do not contain the report heading page, embedded page eject characters, or page or column headings. They consist of an initial page eject followed by a stream of log message text (event type reports) or data (statistics reports). See the CRECAR command in the CDCNET Commmands Reference manual for the COMPRESS parameter usage.

```
REPORT DAY: 88/01/20                                               PAGE     1
                                      EVNTRP1 REPORT
                                   START TIME = 0800 HOURS

  DATE       TIME       SYSTEM ID     LOG ID SEVERITY
======== =========== ===========  ====== ============

88/01/20 08.53.22438 0800253000AE     1554  INFORMATIVE
TELNET terminal device connected to destination: PEWTER
      Local IP address = 192.9.200.15, Local port = 23,
      Remote IP address = 192.9.200.32, Remote port = 1028,
      Device = $CONSOLE_C009C820_0404, Device type = CON,
      TIP = TELNET, Terminal Protocol = NVT,
      Source address = 0000A00F0800253000AEAEA3,
      Destination address = 0000A00F080025D4C0793871,

88/01/20 08.53.36009 0800253000AE     1555  INFORMATIVE
TELNET terminal device disconnected from destination: PEWTER
      Local IP address = 192.9.200.15, Local port = 23,
      ..
```

When generating a report of this format, NPA may encounter a situation where a particular template is not defined. If this happens, the message is formatted using only the data logged by the DI. The first message in the example above would appear as follows:

```
88/01/20 08.53.22438 0800253000AE     1554      ERROR
--ERROR--  CC=0C 3154 TEXT=?PEWTER?192?9?200?15?23?192?9?200?32?1028?
$CONSOLE_C009C820_0404?CON?TELNET?NVT?0000A00F0800253000AEAEA3?
0000A00F080025D4C0793871
```

The common text header is intact and followed by an ERROR indication, product identifier, and template number. The text of the message body consists solely of DI logged data separated by a (?) delimiter.

Expected Operating Limits

Some reports include expected operating limits. This feature allows you to easily detect unsatisfactory performance conditions in your network. The limits are expressed as two numbers; an upper limit and a lower limit, both of which appear in the column heading portion of your report. The column heading occurs on the first column heading line, the upper limit appears on the next line directly below the column title, and the lower limit appears on the following line directly below that. Any values that do not fall within the specified limit range are called to your attention. A less-than symbol (<) appears to the right of a value that is less than the lower limit. A greater-than symbol (>) appears to the right of a value that is greater than the upper limit. These limits are demonstrated in the following example:

```
                                                % BAD                      % BAD
                          BLOCKS    BLOCKS    [   5]  BLOCKS  BLOCKS   [   7]
TIME    CS LI PO SA DA    IN        BAD IN    [   0]   OUT    BAD OUT  [   0]
======  == == == == ==  ========  ========  =======  =======  =======  ======
 0900   01 01 01 23 00       224        22      10>      783       48       5
 0900   01 02 01 24 00       164        78      5        564       28       5
```

In this sample report segment, the limits appear directly beneath the two column headings described as % BAD. The upper number always represents the upper limit; the number directly below always defines the lower limit. Here, the first % BAD field defines as acceptable a limit ranging from 0% to 5%; the second % BAD field specifies a lower limit of 0% and an upper limit of 7%. The value, 10, beneath the first % BAD heading is flagged for your attention because it exceeds the upper limit of 5.

Specific ranges of limits for applicable reports have been defined in your NPA software program. However, you may change existing limits by using the CHAEOL command.

Log Message ID

NPA reports are generated from log messages with unique log message identifiers (ID(s)). Some reports consist of a collection of expanded log message text (configuration report, event message reports, hardware error message reports, and software error message reports). The expanded text for each log message includes the log ID.

The other NPA reports (for example, connection statistics report) are generated from statistical data contained in certain log messages. These reports do not contain the log ID. The log IDs used to generate them are listed in the descriptive text in this manual.

For more information on the IDs, consult the online Diagnostic Messages manual. Or, if you prefer, you can use the EXPCLM command.

Specific Reports

NPA provides you with a standard report set. You can generate some of these report types in more than one format. For example, you can choose from three varieties of reports that provide mainframe channel interface (MCI) statistical information.

The following is a list of NPA report types and the names of the individual reports available for each general report type. This list also contains the report name keyword value that you enter when you execute NPA procedures. The reports are described and illustrated on the pages following the table.

Report Type	Report Name	Keyword
Configuration report	CDCNET Configuration Messages - Sorted by Date and Time	CONFRP1
Connection statistics reports	CDCNET Connection Statistics	CONNRP1
	CDCNET Connection Statistics on a Daily Basis	CONNRP2
Device interface (DI) utilization statistics reports	CDCNET Utilization Statistics - CPU and Memory Utilization	DIOSRP1
	CDCNET Utilization Statistics - CPU and Memory Utilization on a Daily Basis	DIOSRP2
	CDCNET Utilization Statistics - Memory State Utilization	DIOSRP3
	CDCNET Utilization Statistics - Memory Utilization on a Daily Basis	DIOSRP4
Ethernet statistics reports	CDCNET Ethernet Statistics (Number of frames sent through each Ethernet)	ETHRRP1
	CDCNET Ethernet Statistics (Types of errors that occur during transmission)	ETHRRP2
Event log message reports	CDCNET Event Log Messages - Sorted by Date and Time	EVNTRP1
	CDCNET Event Log Messages - Sorted by Severity and DI	EVNTRP2
	CDCNET Event Log Message - Frequency Chart	EVNTRP3
	CDCNET Event Log Message - Frequency Chart Reported by DI	EVNTRP4

Report Type	Report Name	Keyword
High-level data link control (HDLC) interface statistics reports	CDCNET HDLC Statistics Characters, Frames and Messages	HDLCRP1
	CDCNET HDLC Statistics Characters and Messages - Sorted by DI on a Daily Basis	HDLCRP2
	CDCNET HDLC Statistics - Frames and Frame Errors	HDLCRP3
Hardware error message reports	CDCNET Hardware Messages - Sorted by Date and Time	HRDWRP1
	CDCNET Hardware Messages - Sorted by Severity and DI	HRDWRP2
	CDCNET Hardware Messages - Frequency chart	HRDWRP3
	CDCNET Hardware Messages - Frequency chart reported by DI	HRDWRP4
Online loader system statistics report	CDCNET Online Loader System Report	LOADRP1
Mainframe channel statistics reports	CDCNET Mainframe Channel Statistics - Characters and Blocks	MCISRP1
	CDCNET Mainframe Channel Statistics - Sorted by SID on a Daily Basis	MCISRP2
	CDCNET Mainframe Channel Statistics - Block Statistics	MCISRP3
Session statistics report	CDCNET Session Statistics	SESSRP1
Software error message reports	CDCNET Software Messages - Sorted by Date and Time	SFTWRP1
	CDCNET Software Messages - Sorted by Severity and DI	SFTWRP2
	CDCNET Software Message - Frequency Chart	SFTWRP3
	CDCNET Software Message - Frequency Chart Reported by DI	SFTWRP4

Report Type	Report Name	Keyword
TELNET statistics reports	CDCNET TELNET Statistics	TELNRP1
	CDCNET TELNET Statistics on a Daily Basis	TELNRP2
Terminal statistics reports	CDCNET Terminal Statistics	TERMRP1
	CDCNET Terminal Statistics	TERMRP2
User statistics report	CDCNET User Report Unsorted	USERRP1
X.25 connection statistics reports	CDCNET X25 Connection Statistics	X25CRP1
	CDCNET X25 Connection Statistics on a Daily Basis	X25CRP2

Configuration Report (CONFRP1)

This report provides the configuration report, on an hourly status basis, of the DI hardware and software. The configuration reports allow you to observe hardware availability throughout your network. This information can be used to pinpoint weak areas in the network that may require maintenance.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for this report. The log message IDs are 593 through 597 with an attribute of S.

The configuration report is organized by DI for ease of reading.

CONFRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-2. Following the report heading page is a numbered report data page similar to the following example. Table 6-1 lists the report field names and their definitions.

```
REPORT DAY: 88/02/09                                                    PAGE     1
                                      CONFRP1 REPORT
          TITLE = AHD_NDI_100D6D                          SID = 08002510006D

  DATE       TIME       SYSTEM ID    LOG ID
======== =========== ============  ======

88/02/09 05.55.00204 08002510006D     593
DI SYSTEM STATUS
SYSTEM NAME = AHD_NDI_10006D
SYSTEM ADDRESS = 08002510006D(16)
BOOT VERSION NUMBER = 4308(16)
SOFTWARE RELEASE LEVEL = 4308(16)
NUMBER OF TASKS = 23
FREE SMM MEMORY = 248988
PERCENT CPU UTILIZATION = 13
BUFFER STATE = GOOD
MEMORY STATE = GOOD
DATE AND TIME OF LAST RELOAD = 88/02/05  21.22.59
-
BUFFER STATUS
TYPE         TOTAL BUFFERS  AVAILABLE BUFFERS  BUFFER SIZE
DATA         1720           1074               144
DESCRIPTOR   566            545                32
-
SMM MEMORY STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
1048576       248988            132        58148
-
PMM MEMORY STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
131072        24292             5          0
-
MPB RAM STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
16384         2060              2          0
LARGEST SMM MEMORY EXTENT AVAILABLE = 200690
```

Table 6-1. Field Definitions for Configuration Report

Field	Definition
BOOT VERSION NUMBER	Version number of the boot file currently loaded in and running on the DI. Taken from exception list or INITMDI.
BUFFER STATE	Describes level of buffer availability. The four states of buffer availablity are GOOD, FAIR, POOR, and CONGESTED. The boundaries between these states is set during configuration. Each boundary is expressed as percentage of the total resource currently allocated after DI configuration.
BUFFER STATUS	Displays the following information: Total Buffers — Total number of buffers allocated for use by the DI. Available Buffers — Number of allocated buffers that are not currently in use. Buffer Size — Size in bytes of that particular type of buffer.
DATE	The date on which the log occurred.
FREE SMM MEMORY	The amount of system main memory that is currently available for modules to be loaded into.
LOG ID	The log message identification number.
MEMORY STATE	Describes level of memory availability. The four states of memory availablity are GOOD, FAIR, POOR, and CONGESTED. The boundaries between these states is set during configuration. Each boundary is expressed as a percentage of the total resource currently allocated after DI configuration.

(Continued)

Table 6-1. Field Definitions for Configuration Report *(Continued)*

Field	Definition	
MEMORY STATUS (PMM, SMM, MPB)	Displays the following information:	
	Total Memory	Total number of bytes of memory for this DI.
	Available Memory	Number of bytes of memory available for loading modules and allocating structures.
	Extents	Number of memory fragments into which available memory is reached.
	Deloadable Memory	Number of bytes that can be used when a deloadable threshold is reached. Deloadable memory is comprised of modules without an active task.
NUMBER OF TASKS	The number of tasks executing in the DI.	
PERCENT CPU UTILIZATION	The percent of time the CPU is active.	
SEVERITY	The description of the severity of the type of error that has occurred.	
SOFTWARE RELEASE LEVEL	The level at which the software was compiled.	
SYSTEM ADDRESS	A 12-digit hexadecimal system ID associated with the DI.	
SYSTEM ID	The system identification number that identifies the DI.	
SYSTEM NAME	The logical name given to the DI in its configuration file.	
TIME	The DI clock time when the log occurred.	

Connection Statistics Reports (CONNRP1, CONNRP2)

Connection statistics reports allow you to observe the length and location of terminal session usage throughout your network. This information demonstrates, over a period of time, growth areas in your network defined in sessions. You can then anticipate and meet additional network hardware needs.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. The log message IDs are 618 and 620 with an attribute of S.

NPA provides two connection statistics reports:

CONNRP1	Sorts connection statistics by DI and time.
CONNRP2	Sorts connection statistics by service name and DI on a daily basis.

CONNRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-2 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                   PAGE     1
                                  CONNRP1 REPORT
         TITLE = TDI_6F                                 SID = 08002510006F

ENDING                                              C O N N E C T I O N
 TIME           SERVICE NAME             INITIATION TERMINATE  AVG TIME   MAX TIME
====== ============================ ========== ========= ========== ==========

   500  ARH907                                    1         1        675        675
   800  ARH907                                    3         2         80        120
   800  ARH990                                    1         0          0          0
   900  ARH907                                    8         2        540        590
   900  ARH990                                    2         2        835       1350
  1000  ARH907                                   15        11       1061       2895
  1000  ARH990                                    2         2        132        245
  1100  ARH907                                    5         8       2936       5990
  1100  ARH990                                    2         2        147        285
  1200  ARH907                                    8         3        461       1115
  1200  ARH990                                    3         2        120        140
  1300  ARH907                                    5         5       1959       4065
  1300  ARH990                                    2         1         95         95
  1400  ARH907                                   11         6       1974       5990
  1400  ARH990                                    1         0          0          0
  1500  ARH907                                   16        16        556       2180
  1500  ARH990                                    7         7       3436      18245
  1600  ARH907                                   12        13       2250       5945
  1600  ARH990                                    2         2       2610       4170
  1700  ARH907                                   10        16       2889      10240
  1700  ARH990                                    1         1       3560       3560
  1800  ARH907                                    1         2       1050       1945
  1800  ARH990                                    0         2       3685       7250
```

CONNRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a report data page similar to the following example. Table 6-2 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                   PAGE     1
                                    CONNRP2 REPORT
          TITLE = TDI_6F                               SID = 08002510006F

ENDING                                           C O N N E C T I O N
 DATE          SERVICE NAME            INITIATION TERMINATE  AVG TIME   MAX TIME
====== ============================== ========== ========= ========== ==========

860905 ARH907                                 95        85       1724      10240
860905 ARH990                                 23        21       2036      18245
```

Table 6-2. Field Definitions for Connection Statistics Reports

Field	Definition
CONNECTION AVG TIME	The average length, in seconds, of a terminal session established through the DI.
CONNECTION INITIATION	The number of times during the report interval that a terminal connection was initiated through the DI.
CONNECTION TERMINATE	The number of times during the report interval that a terminal connection through the DI was terminated.
ENDING DATE	The last date that the reported statistics were tabulated.
ENDING TIME	The last time that the reported statistics were tabulated.
CONNECTION MAX TIME	The length, in seconds, of the longest terminal session established through the DI.
SERVICE NAME	The name of the service to which the terminal users are connected.

DI Utilization Statistics Reports (DIOSRP1, DIOSRP2, DIOSRP3, DIOSRP4)

DI utilization statistics reports are designed to assist you in the long-term planning of your CDCNET resources. These reports list the utilization of the primary DI resources, which are the DI central processor unit (CPU) and the DI memory. This information allows you to contrast the amount of CDCNET resources used against the amount of resources available. You use this comparison to determine whether the available resources are sufficient or if your network requires additional hardware resources, additional memory, or reconfiguration with additional communication capabilities.

If the reported statistics indicate that the memory available is being overutilized or underutilized during a specific reporting period, examine the line regulation statistics reported during the same interval on the TERMRP1 and TERMRP2 reports. This comparison may help you to determine if specific lines are causing unsatisfactory utilization.

Use the DEFINE_SOURCE_LOG_GROUP and START_PROCESS_METRICS commands to gather the statistics needed for these reports. The log message ID for these reports is 299 with an attribute of S.

NPA provides four DI utilization statistics reports:

DIOSRP1	Provides an hourly device operating report on CPU and memory utilization statistics.
DIOSRP2	Provides a daily device operating report on CPU and memory utilization statistics.
DIOSRP3	Provides an hourly device operating report on memory state transitions statistics.
DIOSRP4	Provides a daily device operating report on memory state transitions statistics.

DIOSRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-3 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                PAGE     1
                                      DIOSRP1 REPORT
       TITLE = TDI_78                                       SID = 080025100078
```

ENDING TIME	MPB-CPU AVG UTL	MPB-CPU MAX UTL / MIN UTL	DTA BUF AVG USE	DTA BUF MAX USE / MIN USE	ALC MEM AVG USE	ALC MEM MAX USE / MIN USE	SMM TOTAL	TAM TOTAL	% TAM IN BUFF
0112	4	14	71	73	58	58	1048576	464826	63
		2		69		58			
0212	4	18	70	72	58	58	1048576	464826	63
		2		69		58			
0312	4	15	70	72	57	58	1048576	464826	63
		2		69		57			
0412	4	14	70	72	57	58	1048576	464826	63
		2		69		57			
0512	4	15	72	75	57	58	1048576	464826	63
		2		70		57			
0612	4	16	71	74	57	58	1048576	464826	63
		2		69		57			
0712	4	15	72	74	57	58	1048576	464826	63
		2		69		57			
0812	5	19	71	76	58	60	1048576	464826	63
		2		65		58			
0912	5	23	71	74	59	61	1048576	464826	63
		2		67		58			
1012	5	20	70	74	60	62	1048576	464826	63
		2		68		60			
1212	5	21	72	75	60	62	1048576	464826	63
		2		68		59			
1312	6	22	72	75	60	62	1048576	464826	63
		2		68		60			
1412	5	21	71	75	59	61	1048576	464826	63
		2		67		58			
1512	6	20	70	75	59	60	1048576	464826	63
		2		66		59			
1612	5	20	71	75	59	60	1048576	464826	63
		2		68		59			
1712	5	23	71	75	58	61	1048576	464826	63
		2		66		58			
1912	4	14	72	75	57	57	1048576	464826	63
		2		70		57			
2012	3	14	71	73	57	57	1048576	464826	63
		2		70		57			
2112	3	14	71	73	57	57	1048576	464826	63
		2		70		57			

DIOSRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-3 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                    PAGE     1
                                      DIOSRP2 REPORT
          TITLE = TDI_78                                    SID = 080025100078

                MPB-CPU          DTA BUF          ALC MEM AVERAGE AVERAGE AVERAGE
ENDING MPB-CPU MAX UTL DTA BUF MAX USE ALC MEM MAX USE   SMM     TAM    % TAM
 DATE  AVG UTL MIN UTL AVG USE MIN USE AVG USE MIN USE  TOTAL   TOTAL  IN BUFF
====== ======= ======= ======= ======= ======= ======= ======= ======= =======


860905       4      23      71      76      58      62   96195  464826      63
                     2              65              57
860906       3      15      71      75      56      57 1048576  464826      63
                     2              70              56
```

DIOSRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-3 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                   PAGE     1
                                 DIOSRP3 REPORT
           TITLE = TDI_A2                                  SID = 0800253000A2
```

ENDING TIME	TAM GOOD TIM/TRAN	TAM FAIR TIM/TRAN	TAM POOR TIM/TRAN	TAM CONG. TIM/TRAN	BUF GOOD TIM/TRAN	BUF FAIR TIM/TRAN	BUF POOR TIM/TRAN	BUF CONG. TIM/TRAN
======	========	========	========	========	========	========	========	========
0105	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0205	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0305	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0405	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0505	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0605	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0705	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0805	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
0905	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1005	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1205	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1305	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1415	3562	0	0	0	3562	0	0	0
	0	0	0	0	0	0	0	0
1515	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1615	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1715	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1815	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
1915	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0
2015	3600	0	0	0	3600	0	0	0
	0	0	0	0	0	0	0	0

DIOSRP4 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-3 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                  PAGE     1
                                        DIOSRP4 REPORT
          TITLE = TDI_78                                    SID = 080025100078

       TAM      TAM      TAM      TAM      BUF      BUF      BUF      BUF
ENDING GOOD     FAIR     POOR     CONG.    GOOD     FAIR     POOR     CONG.
DATE   TIM/TRAN TIM/TRAN TIM/TRAN TIM/TRAN TIM/TRAN TIM/TRAN TIM/TRAN TIM/TRAN
====== ======== ======== ======== ======== ======== ======== ======== ========


860905     3600        0        0        0     3600        0        0        0
              0        0        0        0        0        0        0        0

860906     3600        0        0        0     3600        0        0        0
              0        0        0        0        0        0        0        0
```

Table 6-3. Field Definitions for DI Utilization Statistics Reports

Field	Definition
ALC MEM AVG USE	The percentage of total allocatable memory (TAM) in use during the report interval.
ALC MEM MAX USE	The maximum percentage of TAM in use during the report interval.
ALC MEM MIN USE	The minimum percentage of TAM in use during the report interval.
AVERAGE SMM TOTAL	The average number of bytes of system main memory available in the DI during the report period.
AVERAGE % TAM IN BUFF	The average percentage of TAM made into buffers after all modules have been loaded into the DI and configuration file processing is completed.
AVERAGE TAM TOTAL	The average number of bytes of TAM available after all modules have been loaded into the DI.
BUF STATE (GOOD, FAIR, POOR, CONG.) TIM/TRAN	State the buffers are in. TIM is the number of times that the buffers were in the specified state when sampled every 10 seconds, and TRAN is the number of times that a transition occurred into a state during the report interval.
DTA BUF AVG USE	The average percentage of data buffers in use.
DTA BUF MAX USE	The maximum percentage of data buffers in use during the report interval.
DTA BUF MIN USE	The minimum percentage of data buffers in use during the report interval.

(Continued)

Table 6-3. **Field Definitions for DI Utilization Statistics Reports** *(Continued)*

Field	Definition
ENDING DATE	The last date during which the reported statistics were tabulated.
ENDING TIME	The last time at which the reported statistics were tabulated.
MPB-CPU AVG UTL	The average percentage[1] of time the CPU has spent during the report interval performing master processor board-originated tasks.
MPB-CPU MAX UTL	The highest percentage[1] of time the CPU has spent on master processor board-originated tasks performed by the CPU during the report interval.
MPB-CPU MIN UTL	The lowest percentage[1] of time the CPU has spent on master processor board-originated tasks performed by the CPU during the report interval.
SMM TOTAL	The total amount, in bytes, of system main memory available in the DI.
% TAM IN BUFF	The percentage of TAM made into buffers after all modules have been loaded into the DI.
TAM STATE (GOOD, FAIR, POOR, CONG.) TIM/TRAN	State that memory is in. TIM is the number of times that the memory was in the specified state when sampled every 10 seconds, and TRAN is the number of times that a transition occurred into a state during the report interval.
TAM TOTAL	The amount, in bytes, of TAM available after all modules have been loaded into the DI.

1. MPB-CPU percent use is calculated when sampled every 10 seconds.

Ethernet Statistics Reports (ETHRRP1, ETHRRP2)

Ethernet statistics reports allow you to evaluate the performance of your Ethernet and to correct problems before they become catastrophic. Reported statistics tell you the numbers and sizes of data frames transmitted over your Ethernet, the types of transmission errors that have occurred, and the number of data transmission collisions over the reported time interval.

Use the DEFINE_SOURCE_LOG_GROUP and START_TRUNK_METRICS commands to gather the statistics needed for these reports. The log message ID is 639 with an attribute of S.

NPA provides two Ethernet statistics reports:

ETHRRP1	Ethernet frame size transmission sorted by SID.
ETHRRP2	Ethernet frame transmission errors sorted by SID.

Ethernet statistics messages are forwarded to the network log file after SID initialization is complete.

ETHRRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-4 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                      PAGE     1
                                              ETHRRP1 REPORT
           TITLE = TDI_78                                       SID = 080025100078

           XMIT   255 XMIT   511 XMIT   767 XMIT  1023 XMIT  1279 XMIT  1535 COLLIS-
TIME    CS RECV   255 RECV   511 RECV   767 RECV  1023 RECV  1279 RECV  1535 IONS
====== == ========= ========= ========= ========= ========= ========= =========

 0912   6      2262        52         2         0         2         7         0
               2651       179       105         0        13         0
 1012   6      2196        41         2         0         2         4         2
               2370       458        63         2         6         0
 1212   6      2629        41         2         0         2         5         5
               2730       472       127         5        20         0
 1312   6      4509        61         2         0         2         5         2
               3423      1615        79         1         4         0
 1412   6      1971        48         3         0         1         4         4
               2506       119        62         8        27         0
 1512   6      4179        34         2         0         4         8         0
               4374       311       173         9        22         0
```

ETHRRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading is a numbered report data page similar to the following example. Table 6-4 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                    PAGE     1
                                         ETHRRP2 REPORT
        TITLE = MTI_83                                    SID = 080025100083

                                          CRC      ALIGN.    OVERRUN  RESOURCE  ABNORMAL
                                        ERRORS    ERRORS    ERRORS    ERRORS     LOGIC
ENDING      FRAMES    FRAMES    [     20] [     30] [     40] [    100] [      3]
 TIME  CS RECEIVED     SENT     [      0] [      0] [      0] [      0] [      0]
====== == ========= ========= ========= ========= ========= ========= =========

 0924   6     13144     13581         0         0        20         0         0
 1024   6     13291     13704         0        27         0      103>         0
 1224   6     11549     11782         0         0         0         0         0
 1424   6      8276      8175         0         0       44>         0         0
 1524   6     13735     14154         0       33>         0         0         0
```

Table 6-4. Field Definitions for ESCI Statistics Reports

Field	Definition
ABNORMAL LOGIC	The number of frames for which transmission was aborted because of the following: **Too many collisions** — Occurs when 16 attempts at sending the same block of data fails. **Lost carrier sense** — Occurs when the carrier sense signal is lost on the transmission line. **Transmission underruns** — Occur when a DI can't get data out of memory at a 10-megahertz rate. Usually due to a coding error. **Hardware errors** — Occurs when a hardware error interrupts the transmission of data.
ALIGN. ERRORS	The number of frames received with alignment errors. This occurs when the frame isn't a valid number of octets (255, 511, 767, or 1279). Also occurs when a DI receives an underrun error while transmitting.
COLLISIONS	The number of meetings in the Ethernet line of simultaneously transmitted information packets, resulting in retransmission.
CRC ERRORS	The number of cyclic redundancy check (CRC) bit errors. This usually indicates a hardware problem with the ESCI board, transceiver, or coax cable.
CS	The card slot number identifying the location of the module within the DI. When reporting on IEI (ICA-II Ethernet Interface) statistics, this number is always 0 and meaningless.
ENDING TIME	The time representing the end of the time interval being reported on.
FRAMES RECEIVED	The number of frames of data received.
FRAMES SENT	The number of frames of data transmitted.

(Continued)

Table 6-4. Field Definitions for ESCI Statistics Reports *(Continued)*

Field	Definition
OVERRUN ERRORS	The number of lost received frames of data because of internal transfer bus (ITB) traffic. This usually occurs when the DI received the data faster than it can be put in memory due to buffer fragments.
RESOURCE ERRORS	The number of frames of data lost because of the unavailability of receive buffers due to insufficient memory in the DI.
TIME	The time at which the reported statistics are tabulated.
XMIT 255/RECV 255	The number of frames transmitted with 255 or fewer bytes.
XMIT 511/RECV 511	The number of frames transmitted with fewer than 511 bytes and more than 255.
XMIT 767/RECV 767	The number of frames transmitted with fewer than 767 bytes and more than 511.
XMIT 1033/RECV 1033	The number of frames transmitted with fewer than 1033 bytes and more than 767.
XMIT 1279/RECV 1279	The number of frames transmitted with fewer than 1279 bytes and more than 1033.
XMIT 1535/RECV 1535	The number of frames transmitted with fewer than 1535 bytes and more than 1279.

Event Log Message Reports (EVNTRP1, EVNTRP2, EVNTRP3, EVNTRP4)

Event log message reports encountered in the network by the DI Executive Software Failure Management entity or other software components are reported to the network log file.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. All log message IDs with an attribute of EL are used.

NPA provides four event log messages reports:

EVNTRP1	Lists the network log messages sorted by date and time.
EVNTRP2	Lists the network log messages sorted by severity and DI.
EVNTRP3	Provides a summary of all event log messages by frequency and error severity.
EVNTRP4	Provides a summary of all event log messages by frequency and error severity reported by DI.

EVNTRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-5 lists the report field names and their definitions.

```
REPORT OAY: 86/01/01                                                      PAGE     1
                                        EVNTRP1 REPORT
                                   START TIME = 0000 HOURS

  DATE       TIME       SYSTEM ID     LOG ID SEVERITY
======== =========== ============  ====== ============

86/01/01 00.D0.D0785 0800253000A2     1319    WARNING
--WARNING--  INVALID LOG REQUEST FOR SYSTEM FAILURE TABLE -
INVALID DATA; ENTRY 0 CLEARED

86/01/01 00.00.00787 0800253000A2     1319    WARNING
--WARNING--  INVALID LOG REQUEST FOR SYSTEM FAILURE TABLE -
INVALID DATA; ENTRY 1 CLEARED

86/01/01 00.00.00794 0800253000A2     1319    WARNING
--WARNING--  INVALID LOG REQUEST FOR SYSTEM FAILURE TABLE -
INVALID DATA; ENTRY 5 CLEARED

86/01/01 00.00.00796 0800253000A2     1319    WARNING
--WARNING--  INVALID LOG REQUEST FOR SYSTEM FAILURE TABLE -
INVALID OATA; ENTRY 7 CLEARED

86/01/01 00.00.00950 0800253000A2      608    WARNING
--WARNING--  INITIAL LOAOER CHECKSUM BAD, CHECKSUM = DFFF, DATA BUFFER LENGTH =
 0000FFFF,
DESCRIPTOR BUFFER LENGTH = 0040FFFF.

86/01/01 00.00.01254 0800253000A2      765  INFORMATIVE
DVM HAS STARTED IP NUMBER 0007

86/D1/01 00.00.01316 0800253000A2      475  INFORMATIVE
ETHERNET NETWORK  DEFINED AND STARTEO BY BOOT SOURCE FOR TRUNK .
CARD SLOT = 0007.

86/01/01 00.00.09528 0800253000A2      481  INFORMATIVE
SYSTEM_NAME = TDI_A2
DATA_BUFFER_SIZE = 144
BUFFER_PERCENTAGE = 50
BUFFER_BOUNDARY_PERCENTAGES = (40, 20, 5)
MEMORY_BOUNOARY_PERCENTAGES = (40, 15, 2)
MEMORY_MANAGER_PERIOD = 1
RESERVEO_SYSTEM_SPACE = 1000
STANDARO_STACK_SIZE = 2048
CLOCKING_SYSTEM = NO
```

EVNTRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading is a numbered report data page similar to the following example. Table 6-5 lists the report field names and their definitions.

```
REPORT DAY: 87/03/04                                                  PAGE     1
                                EVNTRP2 REPORT
                                    ERROR

 DATE       TIME        SYSTEM ID     LOG ID SEVERITY
======== ======================== ====== ==============

87/03/04 04.45.38635 D8002510006F     622      ERROR
--ERROR--  PROCEDURE FILE CANNOT BE ACCESSED, FILE = PSUTDP_990
REJECT REASON CODE = 0008

87/03/04 07.41.17809 080025100074      73      ERROR
--ERROR--  DEPENDENT FILE ACCESS RECEIVED AN INVALID UCEPID FROM TRANSPORT.
UCEPID: 001D3DB2.
INDEPENDENT FILE ACCESS PROTOCOL DATA UNIT: <NO DATA>
GENERIC TRANSPORT INDICATION: 0002.

87/03/04 07.26.03909 080025100078      73      ERROR
--ERROR--  DEPENDENT FILE ACCESS RECEIVED AN INVALID UCEPID FROM TRANSPORT.
UCEPID: 001C8D80.
INDEPENDENT FILE ACCESS PROTOCOL DATA UNIT: <NO DATA>
GENERIC TRANSPORT INDICATION: 0002.

87/03/04 04.06.59701 08002510008B     498      ERROR
--ERROR--  K-DISPLAY RECEIVED REQUEST FROM OSA WHEN NOT RUNNING: REQUEST TYPE =
 0000.
     THE REQUEST WILL BE DISCARDED.
     TRUNK NAME = $MCI5.

87/03/04 09.15.34059 08002510008B      73      ERROR
--ERROR--  DEPENDENT FILE ACCESS RECEIVED AN INVALID UCEPID FROM TRANSPORT.
UCEPID: 001ED444.
INDEPENDENT FILE ACCESS PROTOCOL DATA UNIT: <NO DATA>
GENERIC TRANSPORT INDICATION: 0002.

87/03/04 13.24.28648 08002510008B     498      ERROR
--ERROR--  K-DISPLAY RECEIVED REQUEST FROM OSA WHEN NOT RUNNING: REQUEST TYPE =
 0000.
     THE REQUEST WILL BE DISCARDED.
     TRUNK NAME = $MCI5.
```

EVNTRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-5 lists the report field names and their definitions.

REPORT DAY: 86/01/01 PAGE 1

EVNTRP3 REPORT

LOG NUMBER	FREQUENCY	INFORMATIVE	WARNING	ERROR	FATAL	CATASTROPHIC
19	9	9	0	0	0	0
67	2	2	0	0	0	0
129	1	0	0	1	0	0
207	4	0	4	0	0	0
210	1	1	0	0	0	0
429	4	4	0	0	0	0
457	4	0	4	0	0	0
502	3	3	0	0	0	0
546	4	4	0	0	0	0
548	1	1	0	0	0	0
552	9	9	0	0	0	0
559	1	1	0	0	0	0
560	1	1	0	0	0	0
561	1	1	0	0	0	0
575	2	2	0	0	0	0
593	4	4	0	0	0	0
594	4	4	0	0	0	0
595	4	4	0	0	0	0
596	4	4	0	0	0	0
597	4	4	0	0	0	0
603	1	0	0	0	1	0
605	50	50	0	0	0	0
631	1	0	0	1	0	0
TOTALS	119	108	8	2	1	0

EVNTRP4 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-5 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                PAGE    11
                                   EVNTRP4 REPORT
        TITLE = SVLTDI2                                  SID = 0800253000D5

LOG NUMBER FREQUENCY INFORMATIVE WARNING   ERROR   FATAL CATASTROPHIC
========== ========= =========== ======= ======= ======= ============

       207         3           0       3       0       0            0
       593         1           1       0       0       0            0
       594         1           1       0       0       0            0
       595         1           1       0       0       0            0
       596         1           1       0       0       0            0
       597         1           1       0       0       0            0
```

Table 6-5. Field Definitions for Event Log Messages Reports

Field	Definition
DATE	The date on which the error occurred.
FREQUENCY	The number of log messages for this log ID.
INFORMATIVE/WARNING/ ERROR/FATAL/CATASTROPHIC	The number of log messages of this severity. Totals are listed on the last line.
LOG ID	The log message identification number.
LOG NUMBER	The log message identification number.
SEVERITY	The description of the severity of the type of error that has occurred.
SYSTEM ID	The system identification number that identifies the DI.
TIME	The network clock time when the error occurred.

HDLC Interface Statistics Reports (HDLCRP1, HDLCRP2, HDLCRP3)

HDLC interface statistics reports provide character, frames, and message information on a daily or hourly basis. The reports are forwarded to the network log file after DI initialization is completed.

Use the DEFINE_SOURCE_LOG_GROUP and START_TRUNK_METRICS commands to gather the statistics needed for these reports. The log message ID is 665 with an attribute of S.

NPA provides three HDLC interface statistic reports:

HDLCRP1 Provides characters, frames, and messages sorted by the DI on an hourly basis.

HDLCRP2 Provides characters and messages sorted by the DI on a daily basis.

HDLCRP3 Provides S/U frames transmitted/received and error frame information sorted by the DI on an hourly basis.

HDLCRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-6 lists the report field names and their definitions.

REPORT DAY: 86/09/05 PAGE 1

HDLCRP1 REPORT

TITLE = SVLNOI1 SID = 0800253001A7

ENDING TIME	LI	PO	CHARS RECVD	MSGS RECVD	FRAMES DISCAROED	CHARS XMIT	MSGS XMIT	FRAMES RE-XMIT'D
0050	1	0	392390	6118	436	484388	6437	0
0050	1	1	300677	5510	170	562945	5930	0
0050	1	2	0	0	0	0	0	0
0723	1	0	249900	3825	2	270380	4051	0
0723	1	1	219614	3883	1	344911	4151	D
0723	1	2	0	0	0	0	0	0
0823	1	0	320513	4368	31	708292	4914	0
0823	1	1	218909	3984	20	424349	4288	0
D823	1	2	0	D	0	0	0	0
0923	1	0	558019	6268	470	1794165	7655	0
0923	1	1	271817	4780	632	518807	4908	0
0923	1	2	0	0	0	0	0	0
1223	1	0	3372787	13226	27	2070857	121D3	D
1223	1	1	279191	5041	38	574037	5459	0
1223	1	2	0	0	0	0	0	0
1423	1	0	2743135	10081	0	815544	8134	0
1423	1	1	256690	4644	0	464411	4922	0
1423	1	2	0	0	0	0	0	0
1623	1	0	7406521	19878	14	1320397	13978	0
1623	1	1	247386	4367	1	556413	4728	0
1623	1	2	0	0	0	0	0	0
1723	1	0	5220177	15045	1	630203	10597	0
1723	1	1	237856	4110	0	431704	4341	0
1723	1	2	0	0	0	0	0	0
1823	1	0	1275356	6085	0	375556	5470	0
1823	1	1	218100	4379	0	412092	4324	0
1823	1	2	0	0	0	0	0	0
1923	1	0	3588949	11813	0	914496	9537	0
1923	1	1	273157	5012	0	379184	4618	0
1923	1	2	0	0	0	0	0	0
2023	1	0	362115	4335	0	447011	4640	0
2023	1	1	223359	3893	0	361838	3983	0
2023	1	2	0	0	0	0	0	0
2123	1	0	2605898	8818	0	345696	6696	0
2123	1	1	217826	3831	0	394383	4071	0
2123	1	2	0	0	0	0	0	0
2223	1	0	244405	3550	0	307916	3772	0
2223	1	1	216580	3749	0	387946	3958	0
2223	1	2	0	0	0	0	0	0

HDLCRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-6 lists the report names and their definitions.

```
REPORT DAY: 86/09/04                                                  PAGE     1
                                   HDLCRP2 REPORT

                                 CHARS              AVG     CHARS              AVG
ENDING                           RECVD/   MSGS      CHARS/  XMIT/    MSGS      CHARS/
DATE     LI PO       DI          1000     RECVD     MSG     1000     XMIT      MSG
======== == == ============ ======== ======== ====== ======== ======== ======

86/09/04  1  0 0800253001A7      328     4476     73      348     4743     73
86/09/04  1  1 0800253001A7      249     4683     53      469     5024     93
86/09/04  1  2 0800253001A7        0        0      0        0        0      0
86/09/05  1  0 0800253001A7    29669   119329    248    10797   103115    104
86/D9/05  1  1 080D253001A7     3388    60878     55     6185    63563     97
86/09/05  1  2 0800253001A7        0        0      0        0        0      0
86/09/06  1  0 0800253001A7     1852    13812    134     1002    13432     74
86/09/06  1  1 0800253001A7      628    11560     54     1114    12D81     92
86/09/06  1  2 0800253001A7        0        0      0        0        0      0
```

HDLCRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-6 lists the report names and their definitions.

REPORT DAY: 86/09/05 PAGE 1

HOLCRP3 REPORT

TITLE = SVLNDI1 SID = 0800253001A7

ENDING TIME	LI	PO	S FRAMES RECVD	S FRAMES XMIT	U FRAMES RECVD	U FRAMES XMIT	CRC ERRORS	BAD FRAMES	FRAMES OUT OF SEQUENCE
0050	1	0	4208	3845	1	1	313	7	4
0050	1	1	4128	3198	0	0	106	0	2
0050	1	2	0	0	0	0	0	0	0
0723	1	0	2587	2350	1	1	0	0	0
0723	1	1	2705	2300	1	1	0	0	0
0723	1	2	0	0	0	0	0	0	0
0823	1	0	2864	2205	1	2	19	0	2
0823	1	1	3052	2319	0	0	16	0	1
0823	1	2	0	0	0	0	0	0	0
0923	1	0	3417	1370	5	9	306	2	9
0923	1	1	3318	2655	333	332	292	21	7
0923	1	2	0	0	0	0	0	0	0
1223	1	0	1357	2955	3	1	13	0	6
1223	1	1	3952	3197	0	0	27	0	1
1223	1	2	0	0	0	0	0	0	0
1423	1	0	895	3357	0	0	0	0	0
1423	1	1	3341	2961	0	0	0	0	0
1423	1	2	0	0	0	0	0	0	0
1623	1	0	503	8231	4	2	2	0	8
1623	1	1	3239	2510	0	0	0	0	0
1623	1	2	0	0	0	0	0	0	0
1723	1	0	1852	7764	0	0	1	0	0
1723	1	1	2980	2515	0	0	0	0	0
1723	1	2	0	0	0	0	0	0	0
1823	1	0	2347	3568	0	0	0	0	0
1823	1	1	2656	2488	0	0	0	0	0
1823	1	2	0	0	0	0	0	0	0
1923	1	0	2487	6013	0	0	0	0	0
1923	1	1	2776	2794	0	0	0	0	0
1923	1	2	0	0	0	0	0	0	0
2023	1	0	3027	2829	0	0	0	0	0
2023	1	1	2637	2388	0	0	0	0	0
2023	1	2	0	0	0	0	0	0	0
2123	1	0	1930	4927	0	0	0	0	0
2123	1	1	2797	2297	0	0	0	0	0
2123	1	2	0	0	0	0	0	0	0
2223	1	0	2575	2465	0	0	0	0	0
2223	1	1	2730	2300	0	0	0	0	0
2223	1	2	0	0	0	0	0	0	0

Table 6-6. Field Definitions for HDLC Interface Statistics Reports

Field	Definition
AVG CHARS/MSG	The average number of characters per message received from the MCI.
BAD FRAMES	The number of bad frames from the protocol's viewpoint. Bad frames occur when the sequence of numbers is outside the window limits.
CHARS RECVD	The number of characters received at the host from the terminal.
CHARS RECVD/1000	The number of characters, in thousands, received at the host from the terminal.
CHARS XMIT	The number of characters transmitted from the host to the terminal.
CHARS XMIT/1000	The number of characters, in thousands, transmitted from the host to the terminal.
CRC ERRORS	The number of cyclic redundancy check (CRC) bit errors. Usually caused by line noise or too many HDLCs on one CIM.
DI	The device interface identification number.
ENDING DATE	The last date that the reported statistics were tabulated.
ENDING TIME	The clock time representing the end of the time interval being reported on.
FRAMES DISCARDED	The number of frames discarded. Discarded frames are caused by bad frames, CRC errors, and frames out-of-sequence.
FRAMES OUT OF SEQUENCE	The number of frames received out-of-sequence.
FRAMES RE-XMIT'D	The number of discarded frames retransmitted due to bad frames, CRC errors, or frames out-of-sequence.

(Continued)

Table 6-6. Field Definitions for HDLC Interface Statistics Reports *(Continued)*

Field	Definition
LI	The slot number of the line interface module (LIM) on the communications interface module (CIM).
MSGS RECVD	The number of information (I) frames received.
MSGS XMIT	The number of I frames transmitted.
PO	The LIM port number.
S FRAMES RECV	The number of supervision (S) frames received.
S FRAMES XMIT	The number of S frames transmitted.
U FRAMES RECVD	The number of unnumbered (U) frames received.
U FRAMES XMIT	The number of U frames transmitted.

Hardware Error Message Reports (HRDWRP1, HRDWRP2, HRDWRP3, HRDWRP4)

Hardware error reports list the location, nature, and severity of hardware errors occurring in your network. These reports help you to determine which equipment has failed or is about to fail. You may then implement the required maintenance that returns your network to order, or performs the actions that prevent abnormal interruptions of network service.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. All log message IDs with an attribute of HE are used.

NPA offers four hardware errors reports:

HRDWRP1 Provides a chronological listing of hardware errors detected in the network, sorted by the date and time of the error occurrences.

HRDWRP2 Lists hardware errors detected in the network, sorted by DI and error severity.

HRDWRP3 Lists a network log messages frequency chart.

HRDWRP4 Lists a network log messages frequency chart reported by DI.

HRDWRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-7 lists the report field names and their definitions.

```
REPORT DAY: 86/01/01                                                  PAGE     1
                                   HRDWRP1 REPORT
                                START TIME = 0000 HOURS

  DATE       TIME       SYSTEM ID    LOG ID SEVERITY
======== =========== ============  ====== ============

86/01/01 00.00.00927 0800253000A2     338     ERROR
--ERROR--  MPB FAILED ON-BOARD TESTING
BEFORE INITIALIZATION WAS SUCCESSFUL.
SLOT NUMBER= 0
FATAL ERRORS= 7

86/01/01 00.00.00930 0800253000A2     340     ERROR
--ERROR--  PMM HAD RECOVERED PARITY ERRORS
DURING ON-BOARD TESTING.
SLOT NUMBER= 1
ERRORS= 39168
FIRST FAILING ADDRESS= 00010000

86/01/01 00.00.00933 0800253000A2     341     ERROR
--ERROR--  SMM SINGLE BIT ERRORS OCCURRED
DURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 1942
ERROR LOG= 0648

86/01/01 00.00.00935 0800253000A2     342     ERROR
--ERROR--  SMM MULTIPLE BIT ERRORS OCCURRED
DURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 11671
ERROR LOG= 0473

86/01/01 00.00.55028 0800253000A2      19  INFORMATIVE
CONFIGURATION COMPLETE,
CONFIGURATION FILE SOURCE:
NETWORK ID: 41454646, SYSTEM ID: 0800253000BE
```

HRDWRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-7 lists the report field names and their definitions.

```
REPORT DAY: 86/01/01                                                    PAGE     1
                                          HRDWRP2 REPORT
                                              ERROR

 DATE       TIME       SYSTEM ID    LOG ID SEVERITY
======== ======================== ====== ==============

86/01/01 00.00.00927 0800253000A2     338     ERROR
--ERROR--  MPB FAILED ON-BOARD TESTING
BEFORE INITIALIZATION WAS SUCCESSFUL.
SLOT NUMBER= 0
FATAL ERRORS= 7

86/01/01 00.00.00930 0800253000A2     340     ERROR
--ERROR--  PMM HAD RECOVERED PARITY ERRORS
DURING ON-BOARD TESTING.
SLOT NUMBER= 1
ERRORS= 39168
FIRST FAILING ADDRESS= 00010000

86/01/01 00.00.00933 0800253000A2     341     ERROR
--ERROR--  SMM SINGLE BIT ERRORS OCCURRED
DURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 1942
ERROR LOG= 0648

86/01/01 00.00.00935 0800253000A2     342     ERROR
--ERROR--  SMM MULTIPLE BIT ERRORS OCCURRED
DURING INITIALIZATION.
SLOT NUMBER= 2
ERRORS= 11671
ERROR LOG= 0473
```

HRDWRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-7 lists the report field names and their definitions.

```
REPORT DAY: 86/01/01                                                    PAGE   1
                                          HRDWRP3 REPORT

LOG NUMBER FREQUENCY INFORMATIVE WARNING   ERROR   FATAL CATASTROPHIC
========== ========= =========== ======= ======= ======= ============


        19         9           9       0       0       0            0
       351         1           1       0       0       0            0
       457         4           0       4       0       0            0
       578         4           0       0       4       0            0
       631         1           0       0       1       0            0
           --------- ----------- ------- ------- ------- ------------

    TOTALS        19          10       4       5       0            0
```

HRDWRP4 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-7 lists the report field names and their definitions.

```
REPORT DAY: 86/07/08                                                  PAGE   1
                                     HRDWRP4 REPORT
         TITLE = MDI_8C                                      SID = 08002510008C

LOG NUMBER FREQUENCY INFORMATIVE WARNING   ERROR   FATAL CATASTROPHIC
========== ========= =========== ======= ======= ======= ============

       457         3           0       3       0       0            0
       578         3           0       0       3       0            0
       631         1           0       0       1       0            0
```

Table 6-7. Field Definitions for Hardware Error Message Reports

Field	Definition
DATE	The date on which the error occurred.
FREQUENCY	Number of log messages for this log ID.
INFORMATIVE/WARNING/ ERROR/FATAL/CATASTROPHIC	Number of log messages of this severity. Totals are listed on the last line.
LOG ID	The log message identification number.
LOG NUMBER	The log message identification number.
SEVERITY	The description of the severity of the type of error that has occurred.
SYSTEM ID	The system identification number that identifies the DI.
TIME	The network clock time when the error occurred.

Online Loader System Statistics Report (LOADRP1)

Proper module assignment throughout your CDCNET reduces loading overhead and system response times. The loader system statistics report can help you determine the extent of task-loading activities and possible system impact upon network communication between the DI and the source host mainframe.

For example, you may detect that a frequently called module is always being loaded from the host through the network communications link. The resulting network impact of this task can be reduced if its residency assignment is changed from the host to the DI Unused Module List.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for this report. The log message ID is 605 with an attribute of S.

This report chronologically lists the modules that have been loaded in your network.

This report may be produced to cover a daily or weekly time interval.

LOADRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-8 lists the report field names and their definitions.

```
REPORT DAY: 86/06/13                                                  PAGE    1
                                     LOADRP1 REPORT
         TITLE = MTI_1                                    SID = D80025100085

                                             DESTINATION         NO.        CHECK
TIME     MODULE NAME                        MPB  PMM     SMM     LOADS SRCE  SUM
==== ============================== ==== ==== ======== ===== ==== =====

1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0000
1700 ALARMING_COMMAND_PROCESSORS    0000 0000 00002BD8     1 SYS   58AB
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 0EL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 0EL   0000
170D CMD_CHANGE_ELEMENT_STATE       0000 0000 000009F0     1 SYS   732C
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 DD0D0000     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   D000
1700 ACCESS_DIAGNOSTIC_ENTRY        0D0D 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 00000000     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0D00
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 0000000D     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 00000000     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
17DD CMD_CHANGE_ELEMENT_STATE       0000 0000 00000000     1 DEL   D0DD
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 00000000     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     1 DEL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 CMD_CHANGE_ELEMENT_STATE       0000 0000 00000000     1 DEL   0000
1700 OSA_COMMAND_PROCESSORS         0000 0000 00000000     2 0EL   0000
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 DEL   0000
1700 DGMESCO                        0000 0000 000031C8     1 SYS   DD66
1700 DIAGNOSTIC_COMMON_ROUTINES     0000 0000 000005E6     1 SYS   4C19
1700 ACCESS_DIAGNOSTIC_ENTRY        0000 0000 00000000     1 0EL   0000




     SRCE  = SOURCE OF MODULE
     SYS   = SYSTEM LIBRARY
     DEL   = 0ELOADABLE MODULE LIST (I.E. DI MEMORY)
```

Table 6-8. Field Definitions for Online Loader System Statistics Report

Field	Definition
CHECK SUM	This number is a checksum of the loader text. If the module checksum for one module at one version is different, loader text corruption occurs. This leads to unpredictable Dl events. Corruption can occur anywhere in the data path.
DESTINATION MPB	The number of bytes of software information loaded into the main processor board (MPB) of the loaded DI.
DESTINATION PMM	The number of bytes of software information loaded into the private memory module (PMM) of the loaded DI.
DESTINATION SMM	The number of bytes of software information loaded into the system main memory (SMM) of the loaded DI.
MODULE NAME	The name of the loaded module.
NO. LOADS	The number of times the identified module was loaded.
SRCE	The source from which the module is loaded. The source can be SYS (the system library) or DEL (the deloadable module list in the DI memory).
TIME	The time that the reported statistics are tabulated.

Mainframe Channel Statistics Reports (MCISRP1, MCISRP2, MCISRP3)

Mainframe channel statistics reports provide you with information that allows you to monitor the mainframe channel usage in your network, locate mainframe channel transmission problems, and take corrective action that prevents any problems from becoming catastrophic. These reports detail the amount of data transmitted in and out of your mainframe channels for both characters and blocks of data generated and received. MCISRP3 includes the expected operating limits feature, which draws your attention to any unacceptably high frequency of data retransmissions or bad block transmission.

Use the DEFINE_SOURCE_LOG_GROUP and START_TRUNK_METRICS commands to gather the statistics needed for these reports. The log message ID is 562 with an attribute of S.

Three mainframe channel interface reports are provided:

MCISRP1 — Lists the number of characters and blocks of data used as input and output through the mainframe channel. The data presented in MCISRP1 is sorted by the SID.

MCISRP2 — Lists the number of characters and blocks sorted by the SID on a daily basis.

MCISRP3 — Compares the number of good blocks of information to the number of bad blocks of information transmitted. Data is sorted by the SID with EOL.

MCISRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-9 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                PAGE  1
                                   MCISRP1 REPORT
            TITLE = MDI_8A                                   SID = 08002510008A
```

ENDING TIME	CS	CHARS IN	BLOCKS IN	AVERAGE CHAR/BLK	CHARS OUT	BLOCKS OUT	AVERAGE CHAR/BLK
0124	7	0	0	0	0	0	0
0224	7	828	36	23	1203	36	33
0324	7	1656	72	23	2412	72	33
0424	7	0	0	0	0	0	0
0524	7	0	0	0	0	0	0
0624	7	17938	764	23	833851	767	1087
0724	7	13938	280	49	71134	283	251
0824	7	1207006	912	1323	20157	909	22
0924	7	54822	478	114	38290	492	77
1024	7	593878	4419	134	5131138	4431	1158
1224	7	61268	535	114	13446	543	24
1424	7	224311	1032	217	35178	1040	33
1524	7	139976	3328	42	2876477	3332	863
1624	7	51494	2328	22	3145753	2330	1350
1724	7	186372	5029	37	6233287	5041	1236
1824	7	13193	81	162	3181	82	38
1924	7	445	6	74	159	6	26
2024	7	0	0	0	0	0	0
2124	7	0	0	0	0	0	0
2224	7	0	0	0	0	0	0
2324	7	506921	22770	22	30472157	22801	1336

MCISRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-9 lists the report field names and their definitions.

REPORT DAY: 86/09/05 PAGE 1

MCISRP2 REPORT

ENDING DATE	CS	SID	CHARS IN/1000	BLOCKS IN	AVERAGE CHAR/BLK	CHARS OUT/1000	BLOCKS OUT	AVERAGE CHAR/BLK
86/09/05	7	08002510008A	3074	42070	73	48877	42165	1159
86/09/05	4	08002510008B	2627	62549	42	79192	62701	1263
86/09/05	7	080025100081	6593	60372	109	2100	60762	35
86/09/05	7	0800253000BE	86143	247215	348	41464	247987	167
86/09/05	7	0800253000C0	403221	460183	876	23765	461014	52
86/09/06	7	08002510008A	201	9008	22	11875	9027	1316
86/09/06	4	08002510008B	362	14733	25	19227	14758	1303
86/09/06	7	080025100081	471	4511	104	188	4536	42
86/09/06	7	0800253000BE	17404	17186	1013	2849	17204	166
86/09/06	7	0800253000C0	95481	82328	1160	2827	82427	34

MCISRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-9 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                       PAGE      1
                                      MCISRP3 REPORT
          TITLE = MDI_8A                                        SID = 08002510008A

                                      % BAD                              % BAD
 ENDING      BLOCKS   BLOCKS   [      100] BLOCKS   BLOCKS   [      100]
  TIME    CS IN       BAD IN   [        0] OUT      BAD OUT  [        0]
======== == ======== ======== ========= ======== ======== =========

  0124    7        0        0        0        0        0        0
  0224    7       36        0        0       36        0        0
  0324    7       72        0        0       72        0        0
  0424    7        0        0        0        0        0        0
  0524    7        0        0        0        0        0        0
  0624    7      764        0        0      767        0        0
  0724    7      280        0        0      283        0        0
  0824    7      912        0        0      909        0        0
  0924    7      478        0        0      492        0        0
  1024    7     4419        0        0     4431        0        0
  1224    7      535        0        0      543        0        0
  1424    7     1032        0        0     1040        0        0
  1524    7     3328        0        0     3332        0        0
  1624    7     2328        0        0     2330        0        0
  1724    7     5029        0        0     5041        0        0
  1824    7       81        0        0       82        0        0
  1924    7        6        0        0        6        0        0
  2024    7        0        0        0        0        0        0
  2124    7        0        0        0        0        0        0
  2224    7        0        0        0        0        0        0
  2324    7    22770        0        0    22801        0        0
```

Table 6-9. Field Definitions for Mainframe Channel Statistics Reports

Field	Definition
AVERAGE CHAR/BLK (BLOCKS IN)	The average number of characters-per-block received from the mainframe channel or the ICI (ICA-II Channel Interface).
AVERAGE CHAR/BLK (BLOCKS OUT)	The average number of characters-per-block transmitted to the mainframe channel or the ICI.
BLOCKS BAD IN	The number of retransmitted blocks of data received from the mainframe channel or the ICI.
BLOCKS BAD OUT	The number of blocks of data retransmitted to the mainframe channel or the ICI.
BLOCKS IN	The number of data blocks received from the mainframe channel or the ICI.
BLOCKS OUT	The number of data blocks transmitted to the mainframe channel or the ICI.
CHARS IN	The number of characters received from the mainframe channel or the ICI.
CHARS IN/1000	The number of characters, in thousands, received from the mainframe channel or the ICI.
CHARS OUT	The number of characters transmitted to the mainframe channel or the ICI.
CHARS OUT/1000	The number of characters, in thousands, transmitted to the mainframe channel or the ICI.
CS	The card slot number identifying the location of the board within the DI. When reporting on ICI statistics, this number is always 0 and meaningless.
SID	The device interface identification number.
ENDING DATE	The last date that the reported statistics were tabulated.
ENDING TIME	The last time that the reported statistics were tabulated.
% BAD (BLOCKS IN)	The percentage of bad blocks of data received from the mainframe channel.
% BAD (BLOCKS OUT)	The percentage of blocks of data sent to the mainframe channel.

Session Statistics Report (SESSRP1)

The SESSRP1 report logs summary statistics of Session layer activity.

Use the DEFINE_SOURCE_LOG_GROUP and START_PROCESS_METRICS commands to gather the statistics needed for this report. The log message ID is 737 with an attribute of S.

SESSRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-10 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                    PAGE      1
                                        SESSRP1 REPORT
          TITLE = MDI_8A                                    SID = 08002510008A

                      OVERHEAD                  OVERHEAD
ENDING    DATA PDU'S  PDU'S      DATA PDU'S     PDU'S
 TIME     RECEIVED    RECEIVED   TRANSMITTED    TRANSMITTED
======    =========   =========  ============   ============

 0124             0           0             0              0

 0224             0           0             0              0

 0324             0           0             0              0

 0424             0           0             0              0

 0524             0           0             0              0

 0624           646           7            75              4

 0724           140           6            91              3

 0824            47           1           859              2

 0924           146          11           261             15

 1024          3598          13           240              9

 1224           150           5           320              7

 1424           243          35           607              6

 1524          2301          74           860              3

 1624          2193           4           113              2

 1724          4467          18           456             15

 1824            15           1            33              0

 1924             0           0             0              0

 2024             0           0             0              0

 2124             0           0             0              0

 2224             0           0             0              0
```

Table 6-10. Field Definitions for Session Statistics Report

Field	Definition
DATA PDU'S RECEIVED	Number of protocol data units (PDUs) received by the Session layer during the report interval.
DATA PDU'S TRANSMITTED	Number of PDUs transmitted by the Session layer during the report interval.
ENDING TIME	The last time at which the reported statistics are tabulated.
OVERHEAD PDU'S RECEIVED	Number of PDUs received by the Session layer used to control data flow during the report interval.
OVERHEAD PDU'S TRANSMITTED	Number of PDUs transmitted by the Session layer used to control data flow during the report interval.

Software Error Message Reports (SFTWRP1, SFTWRP2, SFTWRP3, SFTWRP4)

Software error reports list the location, nature, and severity of software errors occurring in your network. These reports help you to determine which software routines have failed or are malfunctioning. You may then implement the required maintenance to return your network to order or perform actions to prevent abnormal interruptions of network service.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. All the log message IDs with an attribute of SE are used.

NPA offers four software errors reports:

SFTWRP1 — Provides a chronological listing of software errors detected in the network.

SFTWRP2 — Lists software errors detected in the network sorted by DI and error severity.

SFTWRP3 — Provides a summary of all software errors by frequency and error severity.

SFTWRP4 — Provides a summary of all software errors by frequency and error severity reported by DI.

SFTWRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-11 lists the report field names and their definitions.

```
REPORT DAY: 87/03/04                                                PAGE    1
                                      SFTWRP1 REPORT
                                 START TIME = 0300 HOURS

  DATE       TIME       SYSTEM ID    LOG ID SEVERITY
======== =========== ============ ====== ============

87/03/04 03.17.53759 08002510008B     430      ERROR
--ERROR--  LOG_SUPPORT_APPLICATION RECEIVED A GENERIC_TRANSPORT DISCONNECT
INDICATION
 FOR ALARMING TO SAPID 0000006308002530000D230F6, CONNECTION ESTABLISHMENT WILL
BE RETRIED.
   PEER DISCONNECT REASON = NOT PROVIDED

87/03/04 03.47.14839 0800253000BE    1282      ERROR
--ERROR--  BIP HAS RECEIVED FORWARD DATA FROM APPLICATION UNEXPECTEDLY.
CONNECTION'S BIP TRANSMITTER STATE = TERM PENDING
CONNECTION NUMBER = 0003
COUPLER NODE = 003B.
MFI NODE = 003C.
TRUNK NAME = $MCI7.

87/03/04 03.47.14842 0800253000BE     129      ERROR
--ERROR--  THE NP_IVT GATEWAY MADE A APPL REQUEST TO BIP THAT WAS REJECTED
SVM CEPID = 001065A4
BIP REQUEST = 0002
TRUNK NAME = $MCI7.

87/03/04 03.50.09115 0800253000BE    1282      ERROR
--ERROR--  BIP HAS RECEIVED FORWARD DATA FROM APPLICATION UNEXPECTEDLY.
CONNECTION'S BIP TRANSMITTER STATE = TERM PENDING
CONNECTION NUMBER = 0003
COUPLER NODE = 003B.
MFI NODE = 003C.
TRUNK NAME = $MCI7.

87/03/04 03.50.09117 0800253000BE     129      ERROR
--ERROR--  THE NP_IVT GATEWAY MADE A APPL REQUEST TO BIP THAT WAS REJECTED
SVM CEPID = 0010650C
BIP REQUEST = 0002
TRUNK NAME = $MCI7.
```

SFTWRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-11 lists the report field names and their definitions.

```
REPORT DAY: 87/03/04                                                    PAGE     1
                                          SFTWRP2 REPORT
                                              ERROR

 DATE        TIME       SYSTEM ID     LOG ID SEVERITY
======== ======================== ====== ==============

87/03/04 04.45.16276 08002510D06D      413      ERROR
--ERROR--  LOG_SUPPORT_APPLICATION RECEIVED A GENERIC_TRANSPORT DISCONNECT
INDICATION
 FOR LOGGING TO SAPID 000000010800253000C071BA, CONNECTION ESTABLISHMENT WILL
BE RETRIED.
   PEER DISCONNECT REASON = SERVICE UNAVAILABLE

87/03/04 04.45.15473 08002510006F      413      ERROR
--ERROR--  LOG_SUPPORT_APPLICATION RECEIVED A GENERIC_TRANSPORT DISCONNECT
INDICATION
 FOR LOGGING TO SAPID 41454B4B08002530007091 3E, CONNECTION ESTABLISHMENT WILL
BE RETRIED.
   PEER DISCONNECT REASON = NOT PROVIDED

87/03/04 04.45.39053 08002510006F      430      ERROR
--ERROR--  LOG_SUPPORT_APPLICATION RECEIVED A GENERIC_TRANSPORT DISCONNECT
INDICATION
 FOR ALARMING TO SAPID 0000006308002530000D230F6, CONNECTION ESTABLISHMENT WILL
BE RETRIED.
   PEER DISCONNECT REASON = NOT PROVIDED

87/03/04 07.35.07010 08002510006F       73      ERROR
--ERROR--  DEPENDENT FILE ACCESS RECEIVED AN INVALID UCEPID FROM TRANSPORT.
UCEPID: 001D289E.
INDEPENDENT FILE ACCESS PROTOCOL DATA UNIT: <NO DATA>
GENERIC TRANSPORT INDICATION: 0002.

87/03/04 04.45.18727 080025100074      413      ERROR
--ERROR--  LOG_SUPPORT_APPLICATION RECEIVED A GENERIC_TRANSPORT DISCONNECT
INDICATION
 FOR LOGGING TO SAPID 41454B4B080025300070913E, CONNECTION ESTABLISHMENT WILL
BE RETRIED.
   PEER DISCONNECT REASON = NOT PROVIDED

87/03/04 07.40.53612 080025100074       73      ERROR
--ERROR--  DEPENDENT FILE ACCESS RECEIVED AN INVALID UCEPID FROM TRANSPORT.
UCEPID: 001D31D2.
INDEPENDENT FILE ACCESS PROTOCOL DATA UNIT: <NO DATA>
```

SFTWRP3 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-11 lists the report field names and their definitions.

```
REPORT DAY: 86/01/01                                                    PAGE     1
                                        SFTWRP3 REPORT

LOG NUMBER FREQUENCY INFORMATIVE WARNING   ERROR   FATAL CATASTROPHIC
========== ========= =========== ======= ======= ======= ============

         8         2           2       0       0       0            0
        19         2           2       0       0       0            0
        73        15           0       0      15       0            0
       129        96           0       0      96       0            0
       413        19           0       0      19       0            0
       427         5           5       0       0       0            0
       429         1           1       0       0       0            0
       430         2           0       0       2       0            0
       529         1           0       0       1       0            0
       651      1495           0       0    1495       0            0
       809         2           2       0       0       0            0
      1242         2           0       0       2       0            0
      1257         1           0       0       1       0            0
      1282       139           0       0     139       0            0
           --------- ----------- ------- ------- ------- ------------
    TOTALS      1782          12       0    1770       0            0
```

SFTWRP4 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-3. Following the report heading page is a numbered report data page similar to the following example. Table 6-11 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                PAGE     9
                                  SFTWRP4 REPORT
         TITLE = MDI_300115                          SID = 080025300115

LOG NUMBER FREQUENCY INFORMATIVE WARNING   ERROR   FATAL CATASTROPHIC
========== ========= =========== ======= ======= ======= ============

       413         1           0       0       1       0            0
       429         1           1       0       0       0            0
       430         1           0       0       1       0            0
```

Table 6-11. Field Definitions for Software Error Message Reports

Field	Definition
DATE	The date on which the error occurred.
FREQUENCY	Number of log messages for this log ID.
INFORMATIVE/WARNING/ ERROR/FATAL/CATASTROPHIC	Number of log messages of this severity. Totals are listed on the last line.
LOG ID	The log message identification number.
LOG NUMBER	The log message identification number.
SEVERITY	The description of the severity of the type of error that has occurred.
SYSTEM ID	The system identification number that identifies the DI.
TIME	The network clock time when the error occurred.

TELNET Connection Statistics (TELNRP1, TELNRP2)

TELNET Connection Statistics reports monitor the TELNET usage on your network. These reports show you the number of connections initiated and terminated, average connect time, and maximum connect time.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. The log message IDs are 1554 and 1555 with an attribute of S.

NPA provides the following two TELNET reports:

TELNRP1	Provides an hourly TELNET connection report.
TELNRP2	Provides a daily TELNET connection report.

TELNRP1 Report Example

The first page is an unnumbered report heading page similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-12 lists the report field names and their definitions.

```
REPORT DAY:  88/01/25                                                   PAGE     1
                                         TELNRP1 REPORT
          TITLE = D3000F2                                  SID = 0800253000F2

ENDING                                               C O N N E C T I O N
 TIME              SERVICE NAME            INITIATION TERMINATE  AVG TIME   MAX TIME
====== ============================== ========== ========= ========== ==========

  200   PEWTER                                 10        10          8         10
  600   PEWTER                                  4         4         12         15
  700   PEWTER                                 13        13         10         15
  800   PEWTER                                 14        14         24        215
  900   PEWTER                                 16        14         25        225
 1000   PEWTER                                  3         3        265        760
 1000   PEWTER                                 22        22        145       1170
 1100   PEWTER                                 17        16        251       1160
 1100   PEWTER                                 28        27        102        815
 1200   PEWTER                                  7         6         89        385
 1200   PEWTER                                 16        18         73        365
 1300   PEWTER                                  9         9        405       2915
 1300   PEWTER                                 17        17         81        665
 1400   PEWTER                                  8         7        575       3010
 1400   PEWTER                                  8         7         15         25
 1500   PEWTER                                  8         8        310       1800
 1500   PEWTER                                 17        17         75        855
 1600   PEWTER                                 11        11        674       4970
 1600   PEWTER                                 20        20        221       2775
 1700   PEWTER                                  8         8        733       4130
 1700   PEWTER                                 15        16        200       2865
 1800   PEWTER                                 12        13        115        375
 1800   PEWTER                                 11        11         17         95
 1900   PEWTER                                  7         6        272        595
 1900   PEWTER                                  8         8         21         95
 2000   PEWTER                                  4         5        214        900
 2000   PEWTER                                 14        14         27        230
 2100   PEWTER                                  7         5         58         80
 2100   PEWTER                                 12        12         10         15
 2200   PEWTER                                  4         6        253       1250
 2200   PEWTER                                 12        12         10         15
 2300   PEWTER                                  3         3        120        300
 2300   PEWTER                                 14        14         10         15
 2400   PEWTER                                 17        17         42        295
```

TELNRP2 Report Example

The first page is an unnumbered report heading page similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-12 lists the report field names and their definitions.

```
REPORT DAY:  88/01/25                                                PAGE     1
                                     TELNRP2 REPORT
         TITLE = D3000F2                                 SID = 0800253000F2

ENDING                                          C O N N E C T I O N
 DATE          SERVICE NAME            INITIATION TERMINATE  AVG TIME   MAX TIME
====== =========================== ========== ========= ========== ==========

880125 PEWTER                                108       106        331       4970
880125 PEWTER                                288       287         71       2865
880126 PEWTER                                  4         4         90        220
880126 PEWTER                                 28        28         13        100
```

Table 6-12. Field Definitions for TELNET Statistics Reports

Field	Definition
CONNECTION AVG TIME	The average length, in seconds, of a TELNET session.
CONNECTION INITIATION	The number of times during the report interval that a TELNET connection was initiated.
CONNECTION TERMINATE	The number of times during the report interval that a TELNET connection was terminated.
ENDING DATE	The last date that the reported statistics were tabulated.
ENDING TIME	The last time that the reported statistics were tabulated.
CONNECTION MAX TIME	The length, in seconds, of the longest TELNET session.
SERVICE NAME	The name of the service to which the TELNET users are connected.

Terminal Statistics Reports (TERMRP1, TERMRP2)

Terminal statistics reports monitor terminal usage on your network. These reports show you the amount of input and output information passing to and from your network's terminals. Terminal activity is measured in characters and blocks of information generated and received.

NPA provides two terminal statistics reports. These reports are usually produced on a daily or weekly basis and may be produced to illustrate condensed time periods.

Use the DEFINE_SOURCE_LOG_GROUP and START_LINE_METRICS commands to gather the statistics needed for these reports. The log message ID is 166 with an attribute of S.

NPA has two terminal statistics reports:

TERMRP1 — Lists the number of bad blocks and good blocks of data that are transferred through each CIM, LIM, and DI. The report is sorted by line and terminal. Includes the expected operating limits feature, which calls your attention to problem areas that require corrective action.

TERMRP2 — Lists the characters and blocks of data that are used as input and output through each CIM, LIM, and DI in your network. The information provided in this report is sorted by line and by terminal.

TERMRP1 Report Example

The first page is a unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-13 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                    PAGE     1
                                   TERMRP1 REPORT
              TITLE = TDI_6F                                SID = 08002510006F
```

ENDING TIME	LI	PO	BLOCKS IN	BLOCKS BAD IN	% BAO [3] [0]	BLOCKS OUT	BLOCKS BAD OUT	% BAD [3] [0]	TIME-OUTS
1905	0	0	119	0	0	385	0	0	0
1905	0	1	247	0	0	1015	0	0	0
1905	0	2	350	0	0	1291	0	0	0
1905	0	3	532	0	0	1646	0	0	0
1905	1	0	193	0	0	567	0	0	0
1905	1	1	83	0	0	290	0	0	0
1905	1	2	614	0	0	1990	0	0	0
1905	1	3	494	0	0	1273	0	0	0
1905	2	0	167	0	0	676	0	0	0
1905	2	1	770	0	0	3078	0	0	0
1905	2	2	46	0	0	155	0	0	0
1905	2	3	0	0	0	0	0	0	0
1905	3	0	146	0	0	511	0	0	0
1905	3	1	762	0	0	3480	0	0	0
1905	3	2	695	0	0	1607	0	0	0
1905	3	3	690	0	0	3680	0	0	0
1905	4	0	0	0	0	0	0	0	0
1905	4	1	641	0	0	1904	0	0	0
1905	4	2	0	0	0	0	0	0	0
1905	4	3	0	0	0	0	0	0	0
1905	5	0	0	0	0	0	0	0	0
1905	5	1	39	0	0	171	0	0	0
1905	5	2	725	0	0	2895	0	0	0
1905	5	3	528	0	0	1590	0	0	0
1905	6	0	1164	0	0	4496	0	0	0
1905	6	1	0	0	0	0	0	0	0
1905	6	2	0	0	0	0	0	0	0
1905	6	3	365	0	0	1544	0	0	0
1905	7	0	0	0	0	0	0	0	0
1905	7	1	950	0	0	1994	0	0	0
1905	7	2	818	0	0	3031	0	0	0
1905	7	3	0	0	0	0	0	0	0

TERMRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-13 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                    PAGE     1
                                        TERMRP2 REPORT
       TITLE = TDI_6F                                         SIO = 08002510006F

ENDING              CHARS     CHARS
 TIME     LI PO     IN/K      OUT/K
======    == == ========= =========

 1905      0  0       1.4      47.1
 1905      0  1       1.6     166.3
 1905      0  2       2.1     230.2
 1905      0  3       6.9     182.3
 1905      1  0       2.9      70.8
 1905      1  1       0.7      42.2
 1905      1  2       4.0     297.8
 1905      1  3       1.8     165.9
 1905      2  0       2.0     103.9
 1905      2  1      10.5     668.0
 1905      2  2       0.4      16.8
 1905      2  3       0.0       0.0
 1905      3  0       1.0      69.5
 1905      3  1       4.3     650.9
 1905      3  2       5.2     245.8
 1905      3  3       4.5     704.7
 1905      4  0       0.0       0.0
 1905      4  1       5.1     216.1
 1905      4  2       0.0       0.0
 1905      4  3       0.0       0.0
 1905      5  0       0.0       0.0
 1905      5  1       0.3      27.1
 1905      5  2       8.9     486.7
 1905      5  3       6.6     169.1
 1905      6  0       6.6     734.1
 1905      6  1       0.0       0.0
 1905      6  2       0.0       0.0
 1905      6  3       3.0     257.2
 1905      7  0       0.0       0.0
 1905      7  1       5.8      78.6
 1905      7  2       7.6     444.4
 1905      7  3       0.0       0.0
```

Table 6-13. Field Definitions for Terminal Statistics Reports

Field	Definition
BLOCKS BAD IN	The number of bad blocks of data received at the DI from the terminal.
BLOCKS BAD OUT	The number of bad blocks of data retransmitted from the DI to the terminal.
BLOCKS IN	The number of blocks of data received at the DI from the terminal.
BLOCKS OUT	The number of blocks of data transmitted from the DI to the terminal.
CHARS IN/K	The number of characters, in thousands, received at the DI from the terminal.
CHARS OUT/K	The number of characters, in thousands, transmitted from the DI to the terminal.
ENDING TIME	The clock time representing the end of the time interval being reported on.
LI	The slot number of the LIM on the CIM.
PO	The LIM port number.
TIME-OUTS	The number of time-outs received.
% BAD (BLOCKS IN)	The percentage of bad blocks of data sent from the terminal to the DI.
% BAD (BLOCKS OUT)	The percentage of bad blocks of data sent from the DI to the terminal.

User Statistics Report (USERRP1)

The user statistics report is a debugging tool. It displays the contents of a log message in a readable format; the management data unit (MDU) format. Each variable field is reformatted and identified by the type of field it is. Each field is displayed as *field_type*variable_value. The following is a list of log message fields:

Field Type	Abbreviation	Value Format
Template specification	*TP*	Hexadecimal
Binary octet	*BO*	Hexadecimal
Binary string	*BS*	Binary
Character octet	*CO*	Alphanumeric
Binary integer	*BI*	Decimal
Binary signed integer	*BSI*	Decimal
Binary-coded decimal	*BCD*	Decimal

Any or all log messages in a log file may provide the information needed for this report. The USER parameter in the REFCLF command determines which log messages provide the report information by specifying their corresponding attribute.

USERRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-2. Following the report heading page is a numbered report data page similar to the following example. Table 6-14 lists the report field names and their definitions.

```
REPORT DAY: 86/01/01                                                        PAGE     1
                                 USERRP1 REPORT

  DATE       TIME        DI                  LOG-NUMBER
======== =========== ============            ==========

86/01/01 00.00.00927 0800253000A2            00338
*TP*1030*BI*0*BI*7

86/01/01 00.00.00930 0800253000A2            00340
*TP*1032*BI*1*BI*39168*BO*00010000

86/01/01 00.00.00933 0800253000A2            00341
*TP*1033*BI*2*BI*1942*BO*0648

86/01/01 00.00.00935 0800253000A2            00342
*TP*1034*BI*2*BI*11671*BO*0473

86/01/01 00.00.55028 0800253000A2            00019
*TP*477*TP*480*TP*5029*BO*41454646*BO*0800253000BE

87/03/04 07.28.42691 0800253000A2            00351
*TP*1043*CO*LIM *BO*2608*CO*CIM SLOT*BI*5*CO*LIM SLOT*BI*0*TP*1073*BI*5

87/03/04 07.39.25820 0800253000A2            00019
*TP*477*TP*480*TP*5029*BO*41454646*BO*0800253000BE

87/03/04 07.59.32410 0800253000A2            00351
*TP*1043*CO*CIM *BO*2608*CO*CIM SLOT*BI*5*TP*1073*BI*5
```

Table 6-14. Field Definitions for User Statistics Report

Field	Definition
DATE	The date that the reported statistics are tabulated.
LOG-NUMBER	The identifiable log message number.
DI	The device interface identification number.
TIME	The time that the reported statistics are tabulated.

X.25 Connection Statistics Reports (X25CRP1, X25CRP2)

These reports provide the X.25 connection statistics. These reports are forwarded to the network log file.

Use the DEFINE_SOURCE_LOG_GROUP command to gather the information needed for these reports. The log message IDs are 1160, 1161, 1342, and 1343 with an attribute of S.

There are two X.25 connection statistics reports:

X25CRP1 Provides a connection statistics report sorted by DI and time.

X25CRP2 Provides a connection statistics report sorted by service name and DI on a daily basis.

X25CRP1 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-15 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                              PAGE    1
                                X25CRP1 REPORT
          TITLE = D3000F2                          SID = 0800253000F2
```

ENDING TIME	SERVICE NAME	CONNECTION INITIATION	CONNECTION TERMINATE	AVG TIME	MAX TIME
200	$GW_NP_39	10	10	8	10
600	$GW_NP_39	4	4	12	15
700	$GW_NP_39	13	13	10	15
800	$GW_NP_39	14	14	24	215
900	$GW_NP_39	16	14	25	225
1000		3	3	265	760
1000	$GW_NP_39	22	22	145	1170
1100		17	16	251	1160
1100	$GW_NP_39	28	27	102	815
1200		7	6	89	385
1200	$GW_NP_39	16	18	73	365
1300		9	9	405	2915
1300	$GW_NP_39	17	17	81	665
1400		8	7	575	3010
1400	$GW_NP_39	8	7	15	25
1500		8	8	310	1800
1500	$GW_NP_39	17	17	75	855
1600		11	11	674	4970
1600	$GW_NP_39	20	20	221	2775
1700		8	8	733	4130
1700	$GW_NP_39	15	16	200	2865
1800		12	13	115	375
1800	$GW_NP_39	11	11	17	95
1900		7	6	272	595
1900	$GW_NP_39	8	8	21	95
2000		4	5	214	900
2000	$GW_NP_39	14	14	27	230
2100		7	5	58	80
2100	$GW_NP_39	12	12	10	15
2200		4	6	253	1250
2200	$GW_NP_39	12	12	10	15
2300		3	3	120	300
2300	$GW_NP_39	14	14	10	15
2400	$GW_NP_39	17	17	42	295

X25CRP2 Report Example

The first page is an unnumbered report heading similar to that shown in figure 6-1. Following the report heading page is a numbered report data page similar to the following example. Table 6-15 lists the report field names and their definitions.

```
REPORT DAY: 86/09/05                                                  PAGE     1
                                        X25CRP2 REPORT
           TITLE = D3000F2                              SID = 0800253000F2

ENDING                                          C O N N E C T I O N
 DATE            SERVICE NAME         INITIATION TERMINATE  AVG TIME   MAX TIME
====== ============================== ========== ========= ========== ==========

860905                                       108       106        331       4970
860905 $GW_NP_39                             288       287         71       2865
860906                                         4         4         90        220
860906 $GW_NP_39                              28        28         13        100
```

Table 6-15. Field Definitions for X.25 Connection Statistics Reports

Field	Definition
CONNECTION AVG TIME	The average length in minutes of a terminal session established through the DI.
CONNECTION INITIATION	The number of times during the report interval that a terminal connection was initiated through the DI.
CONNECTION TERMINATE	The number of times during the report interval that a terminal connection initiated through the DI was terminated.
ENDING DATE	The last date during which the reported statistics were tabulated.
ENDING TIME	The clock time representing the end of the time interval reported on.
CONNECTION MAX TIME	The length, in minutes, of the longest terminal session established through the DI.
SERVICE NAME	The name of the service to which the terminal users are connected.

How To Create Customized NPA Reports Using IPF2 Database Files 7

How To Create Customized NPA Reports Using IPF2 Database Files 7

This chapter provides examples of how to create customized NPA reports using the IPF2 database files. If you need more information than is provided with these examples, see the IPF2 Reference manual listed in Additional Related Manuals.

In order to create a customized report using IPF2, you must have the following:

- IPF2 software package including the MONITOR, REPORT, DEFINE, and UTILITY subsystems. The MONITOR subsystem is the central monitoring facility for IPF2. The REPORT subsystem is used to create report programs. The DEFINE subsystem is used to define the format of a database. It is only used if the report being created requires a change in the format of the database being reported (IPF2 requires all variables [virtual fields] used in a report process to be defined in the database format definition). The UTILITY subsystem provides an option for generating a structure listing of the format of the databases. This listing shows the database definitions of all fields in the databases.

- A copy of the NPA database definition files. These files are NPBDBS1, NPBDBS2, NPBDBS3, and if you are using NOS, NPBDBS4. These files are provided to all customers.

Customized Software Error Report Example

In this example, we want to change the standard NPA Software Error Message Report (SFTWRP1 shown in figure 7-1) to do the following:

- Report only catastrophic and fatal errors

- Report the system name instead of system ID

- Sort messages by system name and then date/time

- Produce a different page header

```
REPORT DAY: 88/02/15                                                  PAGE    1
                                   SFTWRP1 REPORT
                               START TIME = 1000 HOURS

  DATE      TIME      SYSTEM ID    LOG ID SEVERITY
======== =========== ===========  ====== ===========

88/02/15 10.13.22861 0800253004F0    1492     ERROR
--ERROR--  TELNET GATEWAY RECEIVED A BAD REQUEST
REQUEST TO = 0002
RETURNED STATUS = 0003

88/02/15 10.15.27786 0800253004F0     366     ERROR
--ERROR--  SESSION LAYER RECEIVED AN UNEXPECTED EVENT FROM A USER
CEPID = 0019B600
STATE = 0009
EVENT = 0004

88/02/15 10.15.27789 0800253004F0    1492     ERROR
--ERROR--  TELNET GATEWAY RECEIVED A BAD REQUEST
REQUEST TO = 0001
RETURNED STATUS = 0004

88/02/15 10.20.10711 0800253004F0    1228  CATASTROPHIC
--CATASTROPHIC--  DI RESET INDICATION.   RESET CODE: 0018
REASON: TASK ERROR WITH NO RECOVERY PROCEDURE
```

Figure 7-1. Software Error Message Report (SFTWRP1)

Step 1

The first step is to produce a listing of the IPF2 NPBSERR database fields (figure 7-2) contained in the NPA database (NPBDBS). To generate this listing, perform the following command sequence (this produces a listing of all the database fields, but this example uses only those fields in file NPBSERR):

NOS Only

1. Set normal mode by entering:

   ```
   NORMAL
   ```

2. Access database structure files by entering:

   ```
   DEFINE,NPBDBS1,NPBDBS2,NPBDBS3,NPBDBS4
   ATTACH,DBS1=NPBDBS1/UN=NETADMN
   ATTACH,DBS2=NPBDBS2/UN=NETADMN
   ATTACH,DBS3=NPBDBS3/UN=NETADMN
   ATTACH,DBS4=NPBDBS4/UN=NETADMN
   COPYEI,DBS1,NPBDBS1
   COPYEI,DBS2,NPBDBS2
   COPYEI,DBS3,NPBDBS3
   COPYEI,DBS4,NPBDBS4
   RETURN,DBS1,DBS2,DBS3,DBS4
   ```

3. Access IPF2 by entering:

   ```
   GET,IPF2/UN=APPLLIB.
   ```

4. Run IPF2 by entering:

   ```
   BEGIN,,IPF2.
   ```

5. The following prompt appears:

   ```
   PLEASE ENTER DATABASE NAME?
   ```

   Enter:

   ```
   NPBDBS
   ```

6. The following prompt appears:

   ```
   USING WHICH VIEW (ENTER A RECORD-NAME OR COMBINATION-NAME)?
   ```

   Enter:

   ```
   carriage return
   ```

7. The following prompt appears:

   ```
   PLEASE ENTER A MONITOR COMMAND, A SUBSYSTEM NAME OR TYPE "HELP".
   ```

   Enter:

   ```
   UTILITY
   ```

8. The following prompt appears:

```
> BEGIN IPF UTILITY
  UTILITY COMMAND?
```

   Enter:

```
LIST
```

9. The following prompt appears:

```
> BEGIN LIST PROCESS
  STRUCTURE LIST OF DATABASE "NPBDBS" MAY BE FOUND ON LOCAL
  FILE "DBLIST"
> END LIST PROCESS
UTILITY COMMAND?
```

   Enter:

```
END
```

10. The following prompt appears:

```
> END IPF UTILITY
MONITOR COMMAND?
```

   Enter:

```
END
```

A listing of the entire definition/structure of the the database (NPBDBS) is now on local file DBLIST and may be printed.

NOS/VE Only

1. Set your working catalog to a permanent catalog by entering:

```
SETWC $USER
```

2. Access the database structure definition files by entering:

```
COPF $SYSTEM.CDCNET.VERSION_xxxx.NPA.NPBDBS1,NPBDBS1
COPF $SYSTEM.CDCNET.VERSION_xxxx.NPA.NPBDBS2,NPBDBS2
COPF $SYSTEM.CDCNET.VERSION_xxxx.NPA.NPBDBS3,NPBDBS3
```

3. Access IPF2 by entering:

```
ATTF $SYSTEM.APPLICATIONS.IPF2.VER_2_6.IPF2
```

4. Run IPF2 by entering:

```
IPF2
```

5. The following prompt appears:

```
PLEASE ENTER DATABASE NAME?
```

   Enter:

```
NPBDBS
```

6. The following prompt appears:

```
USING WHICH VIEW (ENTER A RECORD-NAME OR COMBINATION-NAME)?
```

   Enter:

```
carriage return
```

7. The following prompt appears:

```
PLEASE ENTER A MONITOR COMMAND, A SUBSYSTEM NAME OR TYPE "HELP".
```

   Enter:

```
UTILITY
```

8. The following prompt appears:

```
> BEGIN IPF UTILITY
  UTILITY COMMAND?
```

   Enter:

```
LIST
```

9. The following prompt appears:

```
> BEGIN LIST PROCESS
  STRUCTURE LIST OF DATABASE "NPBDBS" MAY BE FOUND ON LOCAL
  FILE "DBLIST"
> END LIST PROCESS
UTILITY COMMAND?
```

   Enter:

```
END
```

10. The following prompt appears:

```
> END IPF UTILITY
MONITOR COMMAND?
```

   Enter:

```
END
```

A listing of the entire definition/structure of the database (NPBDBS) is now on local file DBLIST and may be printed.

Figure 7-2 shows the listing of NPBSERR database fields created by the sequence in step 1.

```
                         STRUCTURE REPORT OF
                          DATABASE NPBDBS
                             08/12/88

RECORD: NPASER            TYPE: EXTERNAL       FILE(S): NPBSERR
RECORD LENGTH:   1034     ORG: S     RT: Z     NUM: IPF

PH1: DEVICE INTERFACE
PH2: SOFTWARE ERRORS


--------------------      --------      --------   -------       ----
FIELD NAME/               COLUMN        COL POS/   FORMAT/       REQ?
SYNONYM      LEV OCC      HEADINGS      TYPE       PICTURE
--------------------      ------------------------ ------------------

KEY-FIELD                 KEY               1      35X
              1   1       FIELD         KEY          X(035)

STAT-DATE                 DATE              1      6X
              2   1                                  XX/XX/XX

STAT-YEAR                 YEAR              1      2N
              3   1                                  99/

STAT-MONTH                MONTH             3      2N
              3   1                                  99/

STAT-DAY                  DAY               5      2N
              3   1                                  99

STAT-TIME                 TIME              7      9X
              2   1                                  XX/XX/XXXXX

STAT-TMX                  TIME              7      4X
              3   1                                  XXXX

STAT-HOUR                 HOUR              7      2N
              4   1                                  99

STAT-MIN                  MINUTE            9      2N
              4   1                                  99.

STAT-SEC                  SECONDS          11      5N
              3   1                                  99999

NETWORK-ID                NETWORK          16      8X
              2   1       ID                         X(008)
```

Figure 7-2. NPBSERR File

(Continued)

(Continued)

```
                    STRUCTURE REPORT OF
                     DATABASE NPBDBS
                        08/12/88

RECORD: NPASER              TYPE: EXTERNAL      FILE(S): NPBSERR
RECORD LENGTH:   1034       ORG: S     RT: Z    NUM: IPF

PH1: DEVICE INTERFACE
PH2: SOFTWARE ERRORS


--------------------   --------        --------   -------        ----
FIELD NAME/            COLUMN          COL POS/   FORMAT/        REQ?
SYNONYM     LEV OCC    HEADINGS        TYPE       PICTURE
--------------------   -----------------------    ------------------

DI-NUMBER              DI                24       12X
              2   1    NUMBER                       X(012)

SYSTEM-TITLE           SYSTEM            36       31X
              1   1    TITLE                        X(031)

LOG-ID                 LOG               67       5N
              1   1    ID                           -(005)9

CONT-IND               CONTINUE          72       2X
              1   1    INDICATOR                    X(002)

SEVERITY               SEVERITY          74       1X
              1   1                                 X(001)

EXPLANATION            EXPLANATION       75       960X
              1   1                                 X(960)

START-DATE             START                      6X
              1   1    DATE            VIR          XX/XX/XX

END-DATE               END                        6X
              1   1    DATE            VIR          XX/XX/XX

START-TIME             START                      4X
              1   1    TIME            VIR          X(004)

START-HOUR             START                      2N
              2   1    HOUR            VIR          99.

START-MIN              START                      2N
              2   1    MINUTE          VIR          99.
```

Figure 7-2. NPBSERR File

(Continued)

(Continued)

```
                    STRUCTURE REPORT OF
                      DATABASE NPBDBS
                         08/12/88

RECORD: NPASER           TYPE: EXTERNAL       FILE(S): NPBSERR
RECORD LENGTH:   1034    ORG: S     RT: Z     NUM: IPF

PH1: DEVICE INTERFACE
PH2: SOFTWARE ERRORS


--------------------   --------         --------   -------        ----
FIELD NAME/            COLUMN           COL POS/   FORMAT/        REQ?
SYNONYM      LEV OCC   HEADINGS         TYPE       PICTURE
--------------------   ------------------------   -----------------


END-TIME               END                         4X
               1   1   TIME             VIR          X(004)

END-HOUR               END                         2N
               2   1   HOUR             VIR          99.

END-MIN                END                         2N
               2   1   MINUTE           VIR          99.

DI-NO                  DI                          12X
               1   1   NUMBER           VIR          X(012)

SEVERITY-NAME          SEVERITY                    12X
               1   1   NAME             VIR          X(012)

NPA-USER-NAME          USER                        10X
               1   1   NAME             VIR          X(010)

BREAK-IND              BREAK                       7N
               1   1   INDICATOR        VIR          -(007)9

SUBTOTAL                                           7N
               1   1                    VIR          -(007)9

SEVERITY-COUNT                                     7N
               1   5                    VIR          -(007)9

SID-SELECT             SID                         1X
               1   1   SELECT           VIR          X(001)

SID                    SYSTEM                      12X
               1  10   ID               VIR          XXXXXXXXXXXX
```

Figure 7-2. NPBSERR File

(Continued)

(Continued)

```
                    STRUCTURE REPORT OF
                      DATABASE NPBDBS
                         08/12/88


RECORD: NPASER          TYPE: EXTERNAL          FILE(S): NPBSERR
RECORD LENGTH:  1034     ORG: S     RT: Z     NUM: IPF

PH1: DEVICE INTERFACE
PH2: SOFTWARE ERRORS


--------------------   --------       --------  -------        ----
FIELD NAME/            COLUMN         COL POS/  FORMAT/        REQ?
SYNONYM     LEV OCC    HEADINGS       TYPE      PICTURE
--------------------   -------------------------  --------------------

LOGID-SELECT           LOGID                    1X
              1   1    SELECT         VIR         X(001)

ID                     LOG                      5X
              1  10    IDS            VIR         XXXXX

LOGID-EXCLUDE          LOGID                    1X
              1   1    EXCLUDE        VIR         X(001)

EID                    EX LOG                   5X
              1  10    IDS            VIR         XXXXX

SEV                    SEVERITY                 1X
              1   5    LEVELS         VIR         X(001)

SEV-NAME               SEVERITY                 12X
              1   5    NAMES          VIR         XXXXXXXXXXXX

FILL1                  FILLER                   1X
              1   1    ONE            VIR         X(001)

FILL2                  FILLER                   1X
              1   1    TWO            VIR         X(001)

FILL3                  FILLER                   1X
              1   1    THREE          VIR         X(001)

FILL4                  FILLER                   1X
              1   1    FOUR           VIR         X(001)
```

Figure 7-2. NPBSERR File

Step 2

The next step is to define this report based on the database definition. The following monitor commands are used to initiate the IPF2 report session.

```
/DATABASE IS NPBDBS
/VIEW IS NPASER
/REPORT
```

Next, the user is prompted to supply the commands necessary to produce the report. The following produces the customized report output as shown in figure 7-3.

```
************************* COMPONENT NAME: SFTWRP1 *************************
*
*   PURPOSE:
*
*      THIS PROCEDURE WILL GENERATE A SOFTWARE ERROR REPORT SORTED BY SYSTEM
*      NAME, DATE, AND TIME.
*
*   DESIGN:
*
*      EACH EVENT REPORTED TO THE LOG FILE IS LISTED ON THE REPORT.
*      THE DATE, TIME, SYSTEM NAME, ERROR CODE, SEVERITY AND ERROR MESSAGE
*      ARE LISTED.
*
***************************************************************************
SET MARGINS 1,90
SET TERMINAL PRINTER.
SET PAGE-SIZE 60.
SUPPRESS COLUMN HEADINGS.

PAGE-HEADING 1 TAB 5 "NETWORK PERFORMANCE ANALYZER", TAB 66 "RUN DATE: ",
   SPACE 0 DATE.
PAGE-HEADING 2 TAB 5 "VERSION 0123",
   TAB 55 START-DATE, SPACE 1 START-TIME,
   SPACE 1 "-", SPACE 1 END-DATE, SPACE 1 END-TIME.
PAGE-HEADING 3 TAB 5 "CUSTOM REPORT",
   TAB 64 "REPORT DAY: ", SPACE 0 STAT-YEAR, SPACE 0 STAT-MONTH,
   SPACE 0 STAT-DAY.
PAGE-HEADING 5 TAB 26 " CDCNET SOFTWARE MESSAGES".
PAGE-HEADING 6 TAB 28 "SORTED BY DATE AND TIME".
PAGE-HEADING 8 TAB 5 "  DATE", TAB 17 "TIME", TAB 26 " SYSTEM NAME   ",
  TAB 58 "LOG ID", TAB 65 "SEVERITY".
PAGE-HEADING 9 TAB 5 "========", TAB 14 "===========",
  TAB 26 "===============================",
  TAB 58 "======", TAB 65 "============".

PROMPT "ENTER STARTING DATE FOR REPORT (YYMMDD):" FOR START-DATE.
PROMPT "ENTER STARTING TIME FOR REPORT (HHMM):" FOR START-TIME.
PROMPT "ENTER ENDING DATE FOR REPORT (YYMMDD):" FOR END-DATE.
PROMPT "ENTER ENDING TIME FOR REPORT (HHMM):" FOR END-TIME.
PROMPT "ENTER DI NUMBER FOR REPORT (HHHHHHHHHHHH):" FOR DI-NO.

SELECT (((STAT-DATE > START-DATE) AND (STAT-DATE < END-DATE)) OR
       ((STAT-DATE = START-DATE) AND (STAT-TMX >= START-TIME) AND
        (START-DATE <> END-DATE)) OR
```

```
        ((STAT-DATE = END-DATE) AND (STAT-TMX <= END-TIME) AND
         (START-DATE <> END-DATE)) OR
        ((START-DATE = END-DATE) AND (STAT-TMX >= START-TIME) AND
         (STAT-TMX <= END-TIME) AND (STAT-DATE = START-DATE))) AND
         ((DI-NO = DI-NUMBER) OR (DI-NO = "080025000000")) AND
        ((SEVERITY = 4) OR (SEVERITY = 5)).

SORT ON SYSTEM-TITLE, STAT-DATE, STAT-TIME.

SCROLL EXPLANATION 79,12.

LOOKUP SEVERITY-NAME FROM SEVERITY
    AS "INFORMATIVE " FOR "1"
    AS "  WARNING   " FOR "2"
    AS "   ERROR    " FOR "3"
    AS "   FATAL    " FOR "4"
    AS "CATASTROPHIC" FOR "5".

BREAK ON BREAK-IND.

COMPUTE BREAK-IND = LAST BREAK-IND + 1.

IF CONT-IND <> "00"
      MOVE LAST BREAK-IND TO BREAK-IND.

PRINT 1 TAB 5 EXPLANATION.

BREAK-HEADING BREAK-IND 1 TAB 5 STAT-YEAR,
      TAB 8 STAT-MONTH,
      TAB 11 STAT-DAY,
      TAB 14 STAT-HOUR, SPACE 0 ".",
      TAB 17 STAT-MIN,
      TAB 20 STAT-SEC,
      TAB 26 SYSTEM-TITLE,
      TAB 58 LOG-ID,
      TAB 65 SEVERITY-NAME.
GO.
```

When you enter GO, IPF2 generates the report; it then returns you to the REPORT subsystem.

Step 3

The IPF2 SAVE command may be used to save the report program for future execution.

NOTE

The NPA database to be reported on must be a permanent file (direct on NOS) resident in the catalog belonging to the user generating the report. This is a standard IPF2 database residency restriction.

Figure 7-3 provides an example of the report generated with the custom report processor.

```
NETWORK PERFORMANCE ANALYZER                                     RUN DATE:  8/18/88
VERSION 0123                                          00/01/01 0000 - 99/21/31 2400
CUSTOM REPORT                                                  REPORT DAY: 88/02/15

                      CDCNET SOFTWARE MESSAGES
                       SORTED BY DATE AND TIME

  DATE       TIME       SYSTEM NAME                      LOG ID SEVERITY
======== =========== ============================== ====== ============

88/02/15 10.20.10711 AHP_TDI_3004F0                   1228 CATASTROPHIC
--CATASTROPHIC--  DI RESET INDICATION.   RESET CODE: 0018
REASON: TASK ERROR WITH NO RECOVERY PROCEDURE
```

Figure 7-3. Customized Software Error Message Report

Creating a Customized NPA Summary Accounting Statistics Report Example

The NPA CRECAR command does not generate a Summary Accounting Statistics report. To create this report, you must use the IPF2 database files. The following example outlines this procedure.

Step 1

The first step is to produce a listing of the IPF2 NPBSUMM database fields (figure 7-4) contained in the NPA database (NPBDBS). To generate this listing, follow the NOS or NOS/VE customized report example sequence for generating the NPBSERR database field listing described earlier in this chapter.

The following connection statistics are accumulated in the database file NPBSUMM:

Terminal Support (log messages 617-620, 1538)
Telnet TIP (log messages 1462-1466)
X.25 Terminal Gateway (log messages 38, 39)
X.25 Gateway (log messages 1160, 1161)
Passthrough (log messages 235, 236, 239, 240)
Device Outcall (log messages 233,234, 237, 238)

A record is generated for each log message. However some fields in the record are not appropriate for some log messages. Inappropriate fields are set to 0, if numeric, or blank, if alphanumeric.

Figure 7-5 shows the data fields generated for the log messages.

```
                         STRUCTURE REPORT OF
                          DATABASE NPBDBS
                             3/09/90


RECORD: NPASUM          TYPE: EXTERNAL              FILE: NPBSUMM
RECORD LENGTH:    828    ORG: S      RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------        --------    -------        ----
FIELD NAME/             COLUMN          COL POS/    FORMAT/        REQ?
SYNONYM      LEV OCC    HEADINGS        TYPE        PICTURE
--------------------    ------------------------    ------------------
KEY-FIELD               KEY                    1    35X
               1   1    FIELD           KEY           X(035)

STAT-DATE               DATE                   1    6X
               2   1                                  XX/XX/XX

STAT-YEAR               YEAR                   1    2N
               3   1                                  99/

STAT-MONTH              MONTH                  3    2N
               3   1                                  99/

STAT-DAY                DAY                    5    2N
               3   1                                  99

STAT-TIME               TIME                   7    9X
               2   1                                  XX/XX/XXXXX

STAT-TMX                TIME                   7    4X
               3   1                                  XXXX

STAT-HOUR               HOUR                   7    2N
               4   1                                  99.

STAT-MIN                MINUTE                 9    2N
               4   1                                  99.

STAT-SEC                SECONDS               11    5N
               3   1                                  99999

NETWORK-ID              NETWORK               16    8X
               2   1    ID                            X(008)
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                     STRUCTURE REPORT OF
                       DATABASE NPBDBS
                           3/09/90


RECORD: NPASUM          TYPE: EXTERNAL            FILE: NPBSUMM
RECORD LENGTH:    828    ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------        --------   -------      ----
FIELD NAME/             COLUMN          COL POS/   FORMAT/      REQ?
SYNONYM     LEV OCC     HEADINGS        TYPE       PICTURE
--------------------    -----------------------    -------------------
DI-NUMBER               DI                 24      12X
              2   1     NUMBER                       X(012)

SYSTEM-TITLE            SYSTEM             36      31X
              1   1     TITLE                        X(031)

LOG-TYPE                LOG PDU            67      4X
              1   1     TYPE                         X(004)

USER-ID                 USER               71      31X
              1   1     ID                           X(031)

FAMILY-DOMAIN           FAMILY OR         102      31X
              1   1     DOMAIN NAME                  X(031)

APPLICATION-ID          APPLICATION       133      31X
              1   1     ID                           X(031)

CHARS-SENT              CHARACTERS        164      10N
              1   1     SENT                         ZZZZZZZZZ9

CHARS-RECV              CHARACTERS        174      10N
              1   1     RECEIVED                     ZZZZZZZZZ9

BLOCKS-SENT             BLOCKS            184      10N
              1   1     SENT                         ZZZZZZZZZ9

BLOCKS-RECV             BLOCKS            194      10N
              1   1     RECEIVED                     ZZZZZZZZZ9

PROCS-EXEC              PROCEDURES        204      5N
              1   1     EXECUTED                     ZZZZ9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                          DATABASE NPBDBS
                              3/09/90


RECORD: NPASUM          TYPE: EXTERNAL              FILE: NPBSUMM
RECORD LENGTH:    828    ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------        --------    -------       ----
FIELD NAME/             COLUMN          COL POS/    FORMAT/       REQ?
SYNONYM      LEV OCC    HEADINGS        TYPE        PICTURE
--------------------    ------------------------    ------------------
CONNECTION-LENGTH       CONNECTION        209       10N
               1   1    LENGTH                        ZZZZZZZZZ9

CLUSTER-DEVICE          CLUSTER OR        219       7X
               1   1    DEVICE                        X(007)

DEVICE-NAME             DEVICE            226       31X
               1   1    NAME                          X(031)

DEVICE-TYPE             DEVICE            257       3X
               1   1    TYPE                          X(003)

TIP-NAME                TIP               260       31X
               1   1    NAME                          X(031)

TERMINAL-PROTOCOL       TERMINAL          291       10X
               1   1    PROTOCOL                      X(010)

SOURCE-OR-CLIENT-ADD    SOURCE-CLIENT     301       24X
               1   1    ADDRESS                       X(024)

DEST-OR-SERVER-ADD      DEST-SERVER       325       60X
               1   1    ADDRESS                       X(060)

TERMINATION-REASON      TERMINATION       385       31X
               1   1    REASON                        X(031)

LINE-NAME               LINE              416       31X
               1   1    NAME                          X(031)

LINE-SPEED              LINE              447       6N
               1   1    SPEED                         ZZZZZ9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                           DATABASE NPBDBS
                               3/09/90


RECORD: NPASUM          TYPE: EXTERNAL           FILE: NPBSUMM
RECORO LENGTH:   828     ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------   --------        --------   -------        ----
FIELD NAME/            COLUMN          COL POS/   FORMAT/        REQ?
SYNONYM     LEV OCC    HEADINGS        TYPE       PICTURE
--------------------   ------------------------   -------------------
LINE-SU7-TYPE          LINE SUB         453       31X
              1   1    TYPE                         X(031)

LINE-TYPE              LINE             484       3X
              1   1    TYPE                         X(003)

LIM-NUMBER             LIM              487       1N
              1   1    NUMBER                       9

PORT-NUMBER            PORT             488       1N
              1   1    NUMBER                       9

CLUSTER-ADDRESS        CLUSTER          489       2N
              1   1    ADDRESS                      Z9

DEVICE-ADDRESS         DEVICE           491       2N
              1   1    ADDRESS                      Z9

TRUNK-NAME             TRUNK            493       31X
              1   1    NAME                         X(031)

TRUNK-LIM-NUMBER       LIM              524       1N
              1   1    NUMBER                       9

TRUNK-PORT-NUMBER      PORT             525       1N
              1   1    NUMBER                       9

TERMINAL-SPEED         TERMINAL         526       6N
              1   1    SPEED                        ZZZZZ9

PDN-NAME               PDN              532       31X
              1   1    NAME                         X(031)
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                           DATABASE NPBDBS
                               3/09/90


RECORD: NPASUM          TYPE: EXTERNAL           FILE: NPBSUMM
RECORD LENGTH:   828     ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------         --------    -------        ----
FIELD NAME/             COLUMN           COL POS/    FORMAT/        REQ?
SYNONYM      LEV OCC    HEADINGS         TYPE        PICTURE
--------------------    -----------------------      -------------------
DTE-ADDRESS             DTE               563        15X
              1   1     ADDRESS                        X(015)

CIRCUIT-TYPE            CIRCUIT           578        3X
              1   1     TYPE                           X(003)

CHANNEL-NUMBER          CHANNEL           581        6N
              1   1     NUMBER                         ZZZZZ9

SEND-PACK-LEN           SENDING PACKET    587        3N
              1   1     LENGTH-PWR OF 2                ZZ9

RECV-PACK-LEN           RECV PACKET       590        3N
              1   1     LENGTH-PWR OF 2                ZZ9

SEND-WINDOW-SIZE        SENDING           593        3N
              1   1     WINDOW SIZE                    ZZ9

RECV-WINDOW-SIZE        RECEIVING         596        3N
              1   1     WINDOW SIZE                    ZZ9

CALLING-CLASS           CALLING           599        6X
              1   1     CLASS                          X(006)

CALLED-CLASS            CALLED            605        6X
              1   1     CLASS                          X(006)

INITIATOR               INITIATOR         611        6X
              1   1                                    X(006)

CHARGED-SYSTEM          CHARGED           617        3X
              1   1     SYSTEM                         X(003)
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                    STRUCTURE REPORT OF
                     DATABASE NPBDBS
                         3/09/90


RECORD: NPASUM          TYPE: EXTERNAL              FILE: NPBSUMM
RECORD LENGTH:   828     ORG: S      RT: V      NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------        --------    -------         ----
FIELD NAME/             COLUMN          COL POS/    FORMAT/         REQ?
SYNONYM     LEV OCC     HEADINGS        TYPE        PICTURE
--------------------    ------------------------    --------------------
GATEWAY-NAME            GATEWAY          620        31X
             1   1      NAME                          X(031)

DESTINATION-NAME        DESTINATION      651        31X
             1   1      NAME                          X(031)

SERVER-NAME             SERVER           682        31X
             1   1      NAME                          X(031)

CHAR-SENT-APPL          CHARACTERS SENT  713        10N
             1   1      TO APPLICATION                ZZZZZZZZZ9

PACKETS-SENT            PACKETS          723        10N
             1   1      SENT                          ZZZZZZZZZ9

SEGMENTS-SENT           SEGMENTS         733        10N
             1   1      SENT                          ZZZZZZZZZ9

CHAR-RECV-APPL          CHARACTERS RECV  743        10N
             1   1      FROM APPL                     ZZZZZZZZZ9

PACKETS-RECV            PACKETS          753        10N
             1   1      RECEIVED                      ZZZZZZZZZ9

SEGMENTS-RECV           SEGMENTS         763        10N
             1   1      RECEIVED                      ZZZZZZZZZ9

LOCAL-IP-ADDRESS-1      LOCAL IP         773        3N
             1   1      ADDRESS 1                     ZZ9

LOCAL-IP-ADDRESS-2      LOCAL IP         776        3N
             1   1      ADDRESS 2                     ZZ9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                           DATABASE NPBDBS
                              3/09/90


RECORD: NPASUM          TYPE: EXTERNAL              FILE: NPBSUMM
RECORD LENGTH:   828     ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------        --------    -------       ----
FIELD NAME/             COLUMN          COL POS/    FORMAT/       REQ?
SYNONYM     LEV OCC     HEADINGS        TYPE        PICTURE
--------------------    ------------------------    -------------------
LOCAL-IP-ADDRESS-3      LOCAL IP         779        3N
             1   1      ADDRESS 3                     ZZ9

LOCAL-IP-ADDRESS-4      LOCAL IP         782        3N
             1   1      ADDRESS 4                     ZZ9

LOCAL-PORT              LOCAL            785        5N
             1   1      PORT                          ZZZZ9

REMOTE-IP-ADDRESS-1     REMOTE IP        790        3N
             1   1      ADDRESS 1                     ZZ9

REMOTE-IP-ADDRESS-2     REMOTE IP        793        3N
             1   1      ADDRESS 2                     ZZ9

REMOTE-IP-ADDRESS-3     REMOTE IP        796        3N
             1   1      ADDRESS 3                     ZZ9

REMOTE-IP-ADDRESS-4     REMOTE IP        799        3N
             1   1      ADDRESS 4                     ZZ9

REMOTE-PORT             REMOTE           802        5N
             1   1      PORT                          ZZZZ9

CLIENT-LIM              CLIENT           807        1N
             1   1      LIM                           9

CLIENT-PORT             CLIENT           808        1N
             1   1      PORT                          9

SERVER-LIM              SERVER           809        1N
             1   1      LIM                           9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                          DATABASE NPBDBS
                              3/09/90


RECORD: NPASUM          TYPE: EXTERNAL            FILE: NPBSUMM
RECORD LENGTH:    828    ORG: S      RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------   --------        --------   -------        ----
FIELD NAME/            COLUMN          COL POS/   FORMAT/        REQ?
SYNONYM     LEV OCC    HEADINGS        TYPE       PICTURE
--------------------   ------------------------   -------------------
SERVER-PORT            SERVER           810       1N
              1   1    PORT                         9

DIAL-NUMBER            NUMBER TO        811       15X
              1   1    DIAL                         X(015)

DIALING-STATUS         DIALING          826       3X
              1   1    STATUS                       X(003)

PRINT-FLAG-VIR         PRINT                      70X
              1   1    FLAG            VIR          X(D70)

VIR-STAT-DATE          STATISTICS                 6X
              2   1    DATE            VIR          X(006)

VIR-STAT-HOUR          STATISTICS                 2N
              2   1    HOUR            VIR          Z9

VIR-USER-ID            USER                       31X
              2   1    ID              VIR          X(031)

VIR-FAMILY             FAMILY                     31X
              2   1    NAME            VIR          X(031)

LOAD-HOUR-CH           LOAD                       4X
              1   1    HOUR            VIR          X(004)

LOAD-HOUR              LOAD                       4N
              2   1    HOUR            VIR          ZZZ9

SUM-CON-VIR            SUM OF                     10N
              1   1    CONNECTIONS     VIR          ZZZZZZZZZ9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                        STRUCTURE REPORT OF
                          DATABASE NPBDBS
                              3/09/90


RECORD: NPASUM          TYPE: EXTERNAL             FILE: NPBSUMM
RECORD LENGTH:    828    ORG: S      RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------    --------         --------    -------         ----
FIELD NAME/             COLUMN           COL POS/    FORMAT/         REQ?
SYNONYM      LEV OCC    HEADINGS         TYPE        PICTURE
--------------------    ----------------------------    -------------------
SUM-PROCS-VIR           SUM OF                       10N
              1   1     PROCEDURES       VIR           ZZZZZZZZZ9

SERV-CON-VIR            SERVICE                      10N
              1   1     CONNECT TIME     VIR           ZZZZZZZZZ9

NW-CON-VIR              NETWORK                      10N
              1   1     CONNECT TIME     VIR           ZZZZZZZZZ9

RECV-CH-VIR             CHARACTERS                   10N
              1   1     RECEIVED         VIR           ZZZZZZZZZ9

TRANS-CH-VIR            CHARACTERS                   10N
              1   1     TRANSMITTED      VIR           ZZZZZZZZZ9

RECV-BL-VIR             BLOCKS                       10N

              1   1     RECEIVED         VIR           ZZZZZZZZZ9

TRANS-BL-VIR            BLOCKS                       10N
              1   1     TRANSMITTED      VIR           ZZZZZZZZZ9

RECV-CH-APPL-VIR        CHARACTERS                   10N
              1   1     RECV-APPL        VIR           ZZZZZZZZZ9

TRANS-CH-APPL-VIR       CHARACTERS                   10N
              1   1     TRANS-APPL       VIR           ZZZZZZZZZ9

RECV-PACKETS-VIR        PACKETS                      10N
              1   1     RECEIVED         VIR           ZZZZZZZZZ9

TRANS-PACKETS-VIR       PACKETS                      10N
              1   1     TRANSMITTED      VIR           ZZZZZZZZZ9
```

Figure 7-4. NPBSUMM File

(Continued)

(Continued)

```
                         STRUCTURE REPORT OF
                           DATABASE NPBDBS
                              3/09/90


RECORD: NPASUM           TYPE: EXTERNAL            FILE: NPBSUMM
RECORD LENGTH:   828      ORG: S     RT: V     NUM: IPF

PH1: SUMMARY ACCOUNTING
PH2: STATISTICS


--------------------   --------       --------   -------        ----
FIELD NAME/            COLUMN         COL POS/   FORMAT/        REQ?
SYNONYM     LEV OCC    HEADINGS       TYPE       PICTURE
--------------------   ------------------------  -------------------
RECV-SEGMENTS-VIR      SEGMENTS                  10N
             1    1    RECEIVED       VIR          ZZZZZZZZZ9

TRANS-SEGMENTS-VIR     SEGMENTS                  10N
             1    1    TRANSMITTED    VIR          ZZZZZZZZZ9

START-DATE             START                     6X
             1    1    DATE           VIR          XX/XX/XX

END-DATE               END                       6X
             1    1    DATE           VIR          XX/XX/XX

START-TIME             START                     4X
             1    1    TIME           VIR          X(004)

START-HOUR             START                     2N
             2    1    HOUR           VIR          99.

START-MIN              START                     2N
             2    1    MINUTE         VIR          99.

END-TIME               END                       4X
             1    1    TIME           VIR          X(004)

END-HOUR               END                       2N
             2    1    HOUR           VIR          99.

END-MIN                END                       2N
             2    1    MINUTE         VIR          99.
```

Figure 7-4. NPBSUMM File

SUMMARY ACCOUNTING DATABASE (NPBSUMM)

	X.25 Terminal Gateway		Passthrough and Device Outcall								Terminal Support					X 25 Gateway		X 25 Asynctip					TELNET TIP				
DATA FIELD \ LOG MSG.	38	39	233	234	235	236	237	238	239	240	617	618	619	620	1538	1160	1161	1340	1341	1342	1343	1344	1462	1463	1464	1465	1466
KEY-FIELD (HEADER)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SYSTEM-TITLE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LOG-TYPE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
USER-ID	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FAMILY-DOMAIN	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
APPLICATION-ID	X	X	X	X	X	X	X	X	X	X						X	X										
CHARS-SENT		X			X		X	X	X	X			X	X			X	X	X		X			X		X	
CHARS-RECV		X			X		X	X	X	X			X	X			X		X		X			X		X	
BLOCKS-SENT					X		X	X	X	X			X	X			X				X			X		X	
BLOCKS-RECV					X		X	X	X	X			X	X			X				X			X		X	
PROCS-EXEC													X						X					X			
CONNECTION-LENGTH (in seconds)		X			X		X	X	X	X			X	X			X		X		X			X		X	
CLUSTER-DEVICE											X		X		X			X	X			X	X	X			X

1

Figure 7-5. NPBSUMM Log Message/Data Field Associations

(Continued)

1. X represents data value supplied.

(Continued)

2

SUMMARY ACCOUNTING DATABASE (NPBSUMM)

	X.25 Terminal Gateway		Passthrough and Device Outcall								Terminal Support					X.25 Gateway		X.25 Asynctip					TELNET TIP				
* LOG MSG. * DATA FIELD *	38	39	233	234	235	235	237	238	239	240	617	618	619	620	1538	1160	1161	1340	1341	1342	1343	1344	1462	1463	1464	1465	1466
DEVICE-NAME											X	X	X	X	X			X	X	X	X	X	X	X	X	X	X
DEVICE-TYPE											X	X	X	X	X					X	X	X	X	X	X	X	X
TIP-NAME											X	X	X	X	X			X	X	X	X	X	X	X	X	X	X
TERMINAL-PROTOCOL											X	X	X	X	X			X	X	X	X	X	X	X	X	X	X
SOURCE-OR-CLIENT-ADD												X		X		X	X			X	X				X	X	
DEST-OR-SERVER-ADD			X	X	X	X	X	X	X	X		X		X		X	X			X	X				X	X	
TERMINATION-REASON		X											(1)	(1)			X		(1)		(1)			(1)		(1)	
LINE-NAME											X	X	X	X	X												
LINE-SPEED											X	X	X	X	X												
LINE-SUB-TYPE											X	X	X	X	X												
LINE-TYPE											X	X	X	X	X												
LIM-NUMBER											X	X	X	X	X												
PORT-NUMBER											X	X	X	X	X												

Figure 7-5. NPBSUMM Log Message/Data Field Associations

(Continued)

2. X represents data value supplied.

(Continued)

S U M M A R Y   A C C O U N T I N G   D A T A B A S E   (NPBSUMM)

* DATA FIELD * / * LOG MSG *	X.25 Terminal Gateway		Passthrough and Device Outcall								Terminal Support					X.25 Gateway		X.25 Asynctip					TELNET TIP				
	38	39	233	234	235	236	237	238	239	240	617	618	619	620	1538	1160	1161	1340	1341	1342	1343	1344	1462	1463	1464	1465	1466
CLUSTER-ADDRESS											X	X	X	X	X												
DEVICE-ADDRESS											X	X	X	X	X												
TRUNK-NAME	X	X														X	X	X	X	X	X	X					
TRUNK-LIM-NUMBER	X	X														X	X	X	X	X	X	X					
TRUNK-PORT-NUMBER	X	X														X	X	X	X	X	X	X					
TERMINAL-SPEED																		X	X	X	X	X					
PDN-NAME	X	X														X	X	X	X	X	X	X					
DTE-ADDRESS	X	X														X	X	X	X	X	X	X					
CIRCUIT-TYPE																X	X	X	X	X	X	X					
CHANNEL-NUMBER	X	X														X	X			X	X						
SEND-PACK-LEN	X	X														X	X	X	X	X	X	X					
RECV-PACK-LEN	X	X														X	X	X	X	X	X	X					
SEND-WINDOW-SIZE	X	X														X	X	X	X	X	X	X					

3

Figure 7-5. NPBSUMM Log Message/Data Field Associations

(Continued)

3. X represents data value supplied.

(Continued)

SUMMARY ACCOUNTING DATABASE (NPBSUMM)

	X.25 Terminal Gateway		Passthrough and Device Outcall								Terminal Support					X.25 Gateway		X.25 Asynctip					TELNET TIP				
* LOG MSG. * DATA FIELD *	38	39	233	234	235	236	237	238	239	240	617	618	619	620	1538	1160	1161	1340	1341	1342	1343	1344	1462	1463	1464	1465	1466
RECV-WINDOW-SIZE	X	X														X	X	X	X	X	X	X					
CALLING-CLASS	X	X														X	X	X	X	X	X	X					
CALLED-CLASS	X	X														X	X	X	X	X	X	X					
INITIATOR	X	X														X	X										
CHARGED-SYSTEM	X	X														X	X	X	X	X	X	X					
GATEWAY-NAME																X	X										
DESTINATION-NAME												X		X		X	X			X	X				X	X	
SERVER-NAME																							X	X	X	X	X
CHAR-SENT-APPL																	X										
PACKETS-SENT		X															X		X								
SEGMENTS-SENT		X															X		X								
CHAR RECV-APPL																	X										
PACKETS-RECV		X															X		X								

4

Figure 7-5. NPBSUMM Log Message/Data Field Associations
(Continued)

4. X represents data value supplied.

(Continued)

[5]

SUMMARY ACCOUNTING DATABASE (NPBSUMM)

DATA FIELD / LOG MSG.	X.25 Terminal Gateway		Passthrough and Device Outcall								Terminal Support					X.25 Gateway		X.25 Asynctip					TELNET TIP				
	38	39	233	234	235	236	237	238	239	240	617	618	619	620	1538	1160	1161	1340	1341	1342	1343	1344	1462	1463	1464	1465	1466
SEGMENTS-RECV		X															X		X								
LOCAL-IP-ADDRESS (4 fields)																							X	X	X	X	X
LOCAL-PORT																							X	X	X	X	X
REMOTE-IP-ADDRESS (4 fields)																							X	X	X	X	X
REMOTE-PORT																							X	X	X	X	X
CLIENT-LIM					X	X			X	X																	
CLIENT-PORT					X	X			X	X																	
SERVER-LIM			X	X	X	X	X	X	X	X																	
SERVER-PORT			X	X	X	X	X	X	X	X																	
DIAL-NUMBER				X		X		X		X																	
DIALING-STATUS				X		X		X		X																	

(1) INCLUDED IF CHARACTER FIELD - BLANK IF TEMPLATE WAS SPECIFIED

Figure 7-5. NPBSUMM Log Message/Data Field Associations

5. X represents data value supplied.

Step 2

The next step is to define this report based on the database definition. The following monitor commands are used to initiate the IPF2 report session.

```
/DATABASE IS NPBDBS
/VIEW IS NPASUM
/REPORT
```

Next, the user is prompted to supply the commands necessary to produce the report. The following produces the customized report output as shown in figure 7-6.

```
SET MARGINS 1,95
SET TERMINAL PRINTER.
SET PAGE-SIZE 62.
SUPPRESS COLUMN HEADINGS.
SUPPRESS DETAIL.
SUPPRESS TOTAL.
*
REPORT-HEADING 1 TAB 72 DATE.
REPORT-HEADING 14 TAB 32 "C D C N E T".
REPORT-HEADING 15 TAB 24 "NETWORK PERFORMANCE ANALYZER".
REPORT-HEADING 17 TAB 19 "TERMINAL SUPPORT ACCOUNTING STATISTICS".
REPORT-HEADING 18 TAB 24 "SORTED BY USER ID AND FAMILY".
REPORT-HEADING 20 TAB 17 "TIME PERIOD = ",
TAB 31 START-DATE,SPACE 1 START-TIME,
SPACE 1 "-",SPACE 1 END-DATE,SPACE 1 END-TIME.
*
PAGE-HEADING 1 TAB 71 "PAGE ", SPACE 0 PAGE-NUMBER.
PAGE-HEADING 2 TAB 1 "  ".
PAGE-HEADING 3 TAB 21 "TERMINAL SUPPORT ACCOUNTING STATISTICS".
PAGE-HEADING 5 TAB 24 "USER ID = ", SPACE 0 USER-ID.
PAGE-HEADING 6 TAB 24 "FAMILY/DOMAIN = ",SPACE 0 FAMILY-DOMAIN.
*
PAGE-HEADING 8 TAB 4 "ENDING",
TAB 16 "NETWORK",
TAB 28 "NUMBER",
TAB 39 "NUMBER",
TAB 49 "SERVICE",
TAB 60 "RECEIVED",
TAB 70 "TRANSMITTD".
*
PAGE-HEADING 9 TAB 3 "DATE/TIME",
TAB 16 "CONNECT",
TAB 30 "OF",
TAB 41 "OF",
TAB 49 "CONNECT",
TAB 59 "CHARACTERS",
TAB 70 "CHARACTERS".
*
PAGE-HEADING 10 TAB 17 "TIME",
TAB 26 "PROCEDURES",
TAB 38 "CONNECTS",
TAB 50 "TIME",
TAB 61 "BLOCKS",
TAB 72 "BLOCKS".
```

```
*
PAGE-HEADING 11 TAB 1 "=============",
TAB 15 "==========",
TAB 26 "==========",
TAB 37 "==========",
TAB 48 "==========",
TAB 59 "==========",
TAB 70 "==========".
*
PROMPT "ENTER STARTING DATE FOR REPORT (YYMMDD):" FOR START-DATE.
PROMPT "ENTER STARTING TIME FOR REPORT (HHMM):" FOR START-TIME.
PROMPT "ENTER ENDING DATE FOR REPORT (YYMMDD):" FOR END-DATE.
PROMPT "ENTER ENDING TIME FOR REPORT (HHMM):" FOR END-TIME.
*
SELECT (((STAT-DATE > START-DATE) AND (STAT-DATE < END-DATE)) OR
 ((STAT-DATE = START-DATE) AND (STAT-TMX >= START-TIME) AND
  (START-DATE <> END-DATE)) OR
 ((STAT-DATE = END-DATE) AND (STAT-TMX <= END-TIME) AND
  (START-DATE <> END-DATE)) OR
 ((START-DATE = END-DATE) AND (STAT-TMX >= START-TIME) AND
  (STAT-TMX <= END-TIME) AND (STAT-DATE = START-DATE))).
SELECT ((LOG-TYPE = "1538") OR (LOG-TYPE = "618 ") OR (LOG-TYPE = "619 ") OR
  (LOG-TYPE = "620 ")).
*
SORT HERE ON USER-ID,FAMILY-DOMAIN,STAT-DATE,STAT-HOUR,STAT-MIN,STAT-SEC,NETWORK--
ID.
*
SELECT ((KEY-FIELD NOT= LAST KEY-FIELD) OR
  (USER-ID NOT= LAST USER-ID) OR (FAMILY-DOMAIN NOT= LAST FAMILY-DOMAIN)).
*
BREAK ON PRINT-FLAG-VIR.
*
PAGE ON USER-ID,FAMILY-DOMAIN,STAT-DATE.
*
COMPUTE LOAD-HOUR = (STAT-HOUR + 1) * 100.
*
* THESE MOVES SET UP THE BREAK FIELD
*
MOVE STAT-DATE TO VIR-STAT-DATE.
MOVE STAT-HOUR TO VIR-STAT-HOUR.
MOVE USER-ID TO VIR-USER-ID.
MOVE FAMILY-DOMAIN TO VIR-FAMILY.
*
* ALWAYS CARRY FORWARD THE SUBTOTALS.
*
MOVE LAST NW-CON-VIR TO NW-CON-VIR.
MOVE LAST SERV-CON-VIR TO SERV-CON-VIR.
MOVE LAST SUM-CON-VIR TO SUM-CON-VIR.
MOVE LAST SUM-PROCS-VIR TO SUM-PROCS-VIR.
MOVE LAST RECV-CH-VIR TO RECV-CH-VIR.
MOVE LAST TRANS-CH-VIR TO TRANS-CH-VIR.
MOVE LAST RECV-BL-VIR TO RECV-BL-VIR.
MOVE LAST TRANS-BL-VIR TO TRANS-BL-VIR.
*
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO SUM-CON-VIR.
```

```
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO SUM-PROCS-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO NW-CON-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO SERV-CON-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO RECV-CH-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO TRANS-CH-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO RECV-BL-VIR.
IF (PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 0 TO TRANS-BL-VIR.
*
* CALCULATIONS THAT ARE DEPENDENT ON A CONNECTION EVENT.
*
IF ((LOG-TYPE = "1538") OR (LOG-TYPE = "618 ")) AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR)
COMPUTE SUM-CON-VIR = LAST SUM-CON-VIR + 1.
IF ((LOG-TYPE = "1538") OR (LOG-TYPE = "618 ")) AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE 1 TO SUM-CON-VIR.
*
* CALCULATIONS THAT ARE DEPENDENT ON A TERMINATION EVENT.
*
IF ((LOG-TYPE = "619 ") AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR))
COMPUTE SUM-PROCS-VIR = LAST SUM-PROCS-VIR + PROCS-EXEC.
IF ((LOG-TYPE = "619 ") AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR))
MOVE PROCS-EXEC TO SUM-PROCS-VIR.
*
IF ((LOG-TYPE = "619 ") AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR))
COMPUTE NW-CON-VIR = LAST NW-CON-VIR + CONNECTION-LENGTH.
IF ((LOG-TYPE = "619 ") AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR))
MOVE CONNECTION-LENGTH TO NW-CON-VIR.
*
IF ((LOG-TYPE = "620 ") AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR))
COMPUTE SERV-CON-VIR = LAST SERV-CON-VIR + CONNECTION-LENGTH.
IF ((LOG-TYPE = "620 ") AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR))
MOVE CONNECTION-LENGTH TO SERV-CON-VIR.
*
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR)
COMPUTE RECV-CH-VIR = LAST RECV-CH-VIR + CHARS-RECV.
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE CHARS-RECV TO RECV-CH-VIR.
*
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR)
```

```
COMPUTE TRANS-CH-VIR = LAST TRANS-CH-VIR + CHARS-SENT.
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE CHARS-SENT TO TRANS-CH-VIR.
*
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR)
COMPUTE RECV-BL-VIR = LAST RECV-BL-VIR + BLOCKS-RECV.
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE BLOCKS-RECV TO RECV-BL-VIR.
*
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR = LAST PRINT-FLAG-VIR)
COMPUTE TRANS-BL-VIR = LAST TRANS-BL-VIR + BLOCKS-SENT.
IF ((LOG-TYPE = "619 ") OR (LOG-TYPE = "620 ")) AND
(PRINT-FLAG-VIR <> LAST PRINT-FLAG-VIR)
MOVE BLOCKS-SENT TO TRANS-BL-VIR.
*
* THESE SUBTOTALS ARE PRINTED AS THE MAIN INFORMATION IN THIS REPORT.
*
BREAK-FOOTING PRINT-FLAG-VIR 2 TAB 1 STAT-YEAR, SPACE 0 STAT-MONTH,
SPACE 0 STAT-DAY, SPACE 1 LOAD-HOUR,
TAB 15 NW-CON-VIR,
TAB 26 SUM-PROCS-VIR,
TAB 37 SUM-CON-VIR,
TAB 48 SERV-CON-VIR,
TAB 59 RECV-CH-VIR,
TAB 70 TRANS-CH-VIR.
*
BREAK-FOOTING PRINT-FLAG-VIR 3 TAB 59 RECV-BL-VIR,
TAB 70 TRANS-BL-VIR.
GO.
```

Step 3

The IPF2 SAVE command may be used to save the report program for future execution.

NOTE

The NPA database to be reported on must be a permanent file (direct on NOS) resident in the catalog belonging to the user generating the report. This is a standard IPF2 database residency restriction.

Figure 7-6 provides an example of the report generated with the custom report processor.

```
                                                                    3/14/90

                              C D C N E T
                        NETWORK PERFORMANCE ANALYZER

                   TERMINAL SUPPORT ACCOUNTING STATISTICS
                        SORTED BY USER ID AND FAMILY

                 TIME PERIOD = 00/00/00 0000 - 99/12/31 2400


                    TERMINAL SUPPORT ACCOUNTING STATISTICS

                        USER ID = JFC
                        FAMILY/DOMAIN = FIRST_DOMAIN

   ENDING      NETWORK     NUMBER     NUMBER    SERVICE    RECEIVED  TRANSMITTD
  DATE/TIME    CONNECT       OF         OF      CONNECT   CHARACTERS CHARACTERS
                TIME     PROCEDURES  CONNECTS    TIME       BLOCKS     BLOCKS
============ ========= ========== ========= ========= ========= =========


90/03/13  1500      116          0          5        50        151       1242
                                                               22         24

90/03/13  1600        0          0          3       685        648       4309
                                                               71         73

90/03/13  1700      200          0          4        15        210         43
                                                               16          5
```

Figure 7-6. Customized Terminal Support Accounting Statistics Report

Device Interface Dump Analyzer 8

Device Interface Dump Analyzer 8

The DI Dump Analyzer program resides on a Control Data host computer and runs under NOS/VE or NOS. It processes subcommands that extract and format the information collected when DI memory is written to a dump file. The Dump Analyzer helps troubleshoot CDCNET by identifying events that have caused its DIs to reset.

This chapter includes the following main topics:

- How to Initiate the Dump Analyzer
- Dump Analyzer Conventions
- How to Retrieve a DI Dump File
- How to Use the Dump Analyzer Input File
- How to Manage Dump Analyzer Output
- How To End a Dump Analyzer Session
- How to Transfer Dump Files Between NOS/VE and NOS
- Sample Input File for NOS/VE or NOS
- Sample Output File for NOS/VE or NOS
- Summary of ANACD Subcommands

NOTE

For more detailed information on the commands used in this chapter, see the CDCNET Commands Reference manual.

If you are doing operations on a CDCNET Network Management Station, refer to the CDCNET Network Management Station manual.

How To Initiate the Dump Analyzer

Use one of the following procedures to initiate the Dump Analyzer.

NOS/VE Only

1. Log in to the host computer by entering your required user name and password. If you successfully log in, you receive a slash.

   ```
   /
   ```

2. You use the CREATE_COMMAND_LIST_ENTRY command to make the Dump Analyzer available to your job.

   ```
   crecle $system.cdcnet.version_independent.command_library
   ```

   NOTE

   This command may be added to your user prolog so that the Dump Analyzer is available to you whenever you log in.

3. Execute the ANALYZE_CDCNET_DUMP command, abbreviated ANACD. The output from this command is a summary of the ANACD subcommand descriptions found later in this chapter.

   This example accepts subcommands from file CMDFILE and writes output to the terminal.

   ```
   ANACD,DF=DIAA132,I=CMDFILE
   ```

   This example begins an interactive Dump Analyzer session on a dump file from SYSTEM_0800253000DA.

   ```
   ANACD,DF=$SYSTEM.CDCNET.DUMP.SYSTEM_0800253000DA.FULL_8701050856249189
   ```

   NOTE

   Refer to Dump Analyzer Conventions, later in this chapter, for more information on the ANACD command.

4. You have successfully started a Dump Analyzer session under NOS/VE if you see the Dump Analyzer's banner message and the NOS/VE Dump Analyzer prompt:

   ```
   COPYRIGHT CONTROL DATA CORPORATION 1985, 1986, 1987   ALL RIGHTS RESERVED

   CDCNET DUMP ANALYZER VERSION = 1614

   CDCNET DI SOFTWARE BOOT VERSION RECORDED IN MPB_RAM = 1614

   CDCNET DI SOFTWARE RELEASE LEVEL RECORDED IN SYSTEM_DATA = 1614

   DA/
   ```

NOS Only

1. Log in to the host computer by entering your required user name and password. If you successfully log in, you receive a slash.

   ```
   /
   ```

2. Execute the ANALYZE_CDCNET_DUMP command, abbreviated ANACD. The output from this command is a summary of the ANACD subcommand descriptions found later in this chapter.

   ```
   ANACD,DF=DIAA132,I=CMDFILE
   ```

   This example begins an interactive Dump Analyzer session, specifying 2404 as the version of the Dump Analyzer to be used.

   ```
   ANACD,DF=DSA9189,V=2404
   ```

   NOTE

   Refer to Dump Analyzer Conventions, later in this chapter, for more information on the ANACD command.

3. You have successfully started a Dump Analyzer session under NOS if you see the Dump Analyzer's banner message and the following prompt.

   ```
   COPYRIGHT CONTROL DATA CORPORATION 1985, 1986, 1987   ALL RIGHTS RESERVED

   CDCNET DUMP ANALYZER VERSION = 1614

   CDCNET DI SOFTWARE BOOT VERSION RECORDED IN MPB_RAM = 1614

   CDCNET DI SOFTWARE RELEASE LEVEL RECORDED IN SYSTEM_DATA = 1614

   ?
   ```

Dump Analyzer Conventions

In order to function properly, the Dump Analyzer must be able to locate and interpret certain data structures from the DI dump file. Because these data structures can change (in location or structure) from version to version, it is important to use a Dump Analyzer that can read the dump file under analysis. If you do not specify an alternative version of the Dump Analyzer, the version used is the version selected by your CDCNET site administrator.

To help identify which alternative Dump Analyzer version to use, two version numbers are displayed at the start of the Dump Analyzer session. One identifies the official **release level** of the CDCNET software product. This version number is stored into the DI's System Data Table during initialization.

The other version number, the **boot version**, is stored in MPB RAM and identifies the software version of the boot file that is used to reload the DI. Unless your site develops software in conjunction with Control Data, the boot version number should match the release level version number.

If the version level of the Dump Analyzer you are using does not match the release level version of the dump file, the Dump Analyzer displays diagnostic message number 86 (see appendix J). You might find it necessary to restart the Dump Analyzer program and specify the CDCNET software version level of the dump file for the VERSION (V) parameter on the ANACD command. If a copy of the Dump Analyzer program built at this software version level is available at the site, it is used.

The following conventions are used for the Dump Analyzer.

- Under NOS/VE, the ANACD command and its parameters may be entered in their full or abbreviated forms. Under NOS, only the abbreviated form of this command, including its parameters, is allowed.
- The command name and its parameters may be separated by commas or spaces.
- ANACD may be entered at any time in response to the NOS prompt. However, because some Dump Analyzer displays are in uppercase and lowercase letters, you must be in ASCII mode to receive correctly formatted output. To set ASCII mode at your terminal, simply enter ASCII in response to the NOS prompt before starting your Dump Analyzer session.
- Dump Analyzer messages received using NOS are written to file OUTPUT.
- Under NOS/VE, Dump Analyzer error messages are always written to the file $ERRORS, which is usually attached to your terminal.
- Under both NOS/VE and NOS, error messages are also written to the currently active output file, if this is not a terminal.
- The file named in an output parameter on a subcommand takes precedence over the output file named on the ANACD command during execution of that subcommand.

How To Retrieve a DI Dump File

NOS/VE Only

Under NOS/VE, DI dump files are placed in the SYSTEM_ssssssssssss subcatalog of the $SYSTEM.CDCNET.DUMP catalog, where ssssssssssss represents the system identifier of the source system. If this catalog does not exist when a dump file belonging there is created, it is created by the host at this time.

The file name of a cataloged DI dump file has the form:

```
FULL_yymmddhhmmss
```

where yymmddhhmmss is the timestamp put on the dump file by the DI from which it originated. The complete path to a DI dump file has the following form:

```
$SYSTEM.CDCNET.DUMP.SYSTEM_ssssssssssss.FULL_yymmddhhmmss
```

A dump file can be copied to a file in your local, working, or personal catalog with the SCL COPY_FILE (COPF) command. It is often useful to copy a dump file into a working catalog file named DUMPFIL, since this is the default file name for the DUMP_FILE parameter on the ANACD command.

The following command copies a dump file into file DUMPFIL in the current working catalog. A subsequent ANACD command automatically chooses this dump file for analysis if no other file is specified with the DUMP_FILE parameter.

```
COPF $SYSTEM.CDCNET.DUMP.SYSTEM_0800253000A0.FULL_870216073930 DUMPFIL
```

This dump file was created when the DI with a system identifier of 0800253000A0 reset at about 7:39 a.m. on the 16th of February 1987.

As an alternative to copying the dump file, you can specify its complete path name for the DUMP_FILE parameter on the ANACD command.

NOS Only

Under NOS, each DI dump stored has a NOS permanent file name that is cataloged under user name NETOPS. Access privilege to user name NETOPS, and the dump files stored there, is site-dependent.

The NOS permanent file name for a DI dump file stored under NETOPS is constructed to indicate the sequential position of this dump with respect to other dumps received by the host, as follows:

```
Dyxxnnn
```

Parameter	Description
y	One alphabetic character in the range I through R for dumps from any Mainframe Terminal Interface (MTI), or from the Mainframe DI (MDI) attached to the Control Data host that loads the CDCNET. This character is initially I and is incremented each time the xx field goes from 99 to AA (for example, DI99nnn, DJAAnnn, DJABnnn, and so on). Or, one alphanumeric character in the range S through Z, then 0 through 9, for dumps from DIs other than the MDI attached to the Control Data host that loads the CDCNET. This character is initially S and is incremented each time the xx field goes from 99 back to AA (for example, DS99nnn, DTAAnnn, DTABnnn, and so on).
xx	Two alphanumeric characters in the range AA through ZZ, then 00 through 99. These characters are initially AA and are incremented each time a NOS file is created for a CDCNET dump.
nnn	The three-digit network invocation number that is incremented each time NAM initiated.

The dump file you specify on the ANACD command must be local to your NOS job. Because the ANACD command uses a local file named DUMPFIL if the DF parameter is not specified, it can be useful to attach a local file named DUMPFIL that references the direct-access dump file you are interested in.

For example, the following NOS command creates a local working copy of DIAA132 (which resides under the site-selected user name NETMGR):

```
ATTACH,DUMPFIL=DIAA132/UN=NETMGR
```

CDCNET File Name

All DI dump files are initially created as CDCNET files and are given CDCNET file names that indicate the originating DI's system identifier and the time at reset. CDCNET file names cannot be used on the ANACD command, but they can be very useful for identifying the origin of a dump file.

The CDCNET file name for a DI dump file is constructed as follows:

```
DUMP#FULL_ssssssssssss_yymmddhhmmss
```

Parameter	Description
ssssssssssss	The 12-character ASCII-coded hexadecimal number representing the system identifier of the device interface whose memory has dumped.
yymmddhhmmss	The ASCII-coded hexadecimal number indicating the year, month, day, hour, minute, and second at the time of the dump.

CDCNET-to-NOS File Name Map

A mapping between DI dump file's CDCNET file names and their corresponding NOS file names is retained in NETDIR. NETDIR is a private, direct-access, permanent file that resides under user name NETOPS. NETDIR can be accessed through the Network File Management (NETFM) utility by site-specified users. If you are permitted to access NETDIR, take the following steps to display CDCNET-to-NOS dump file name maps:

1. After you have logged on to the NOS host under a user name with NETDIR access privilege, use the following command to start NETFM:

   ```
   NETFM,UN=NETOPS
   ```

   The UN parameter tells NETFM where to find NETDIR (UN=NETOPS is actually the default username for the NETFM command). You are prompted for NETFM directives with a question mark (?).

2. Enter NETFM's LIST directive to see the directory entries you are interested in. For example, the following LIST directive lists the CDCNET and NOS file names of all dumps from the DI system with a system identifier of 080025100088:

   ```
   LIST,NF=DUMP#FULL_080025100088*,LO=F
   ```

   The NF parameter identifies the CDCNET file name(s) to be searched for in the directory; the asterisk is a wildcard character that represents any string of characters, and which, if entered as the last character of the string, produces a match with any file name that begins with the specified string.

   If you use a list option of full (LO=F), you get a display that includes NOS permanent file names. Otherwise, you see only the CDCNET file names. A full display is formatted as follows:

```
            NETWORK FILE NAME  PFN      TYPE DN  UN        PN      CREATION  LAST MOD
                                      FS                   RT      DATE/TIME DATE/TIME
                               COMMENT


LIST,NF=DUMP#FULL_080025100088*,LO=F

              DUMP#FULL_08002  DIAA613  DIR  2  NETOPS  0        87/02/19. 87/02/19.
              5100088_8702191     3471                  F        17.34.33. 17.34.33.
              73527

              DUMP#FULL_08002  DIAD617  DIR  3  NETOPS  0        87/02/23. 87/02/23.
              5100088_8702231     3857                  F        14.12.55. 14.12.55.
              41348

              DUMP#FULL_08002  DIAE617  DIR  3  NETOPS  0        87/02/23. 87/02/23.
              5100088_8702231     3511                  F        14.22.31. 14.22.31.
              42322
```

The CDCNET file names are in the first column. NOS permanent file names are in the second column. For example, the dump file with CDCNET file name DUMP#FULL_080025100088_870219173527 is stored in a NOS permanent file named DIAA613.

3. Leave NETFM by entering a blank command line (use a carriage return or line feed in response to the NETFM prompt).

See the CDCNET Configuration Guide for more information on using NETFM and its LIST directive.

NOTE

Under NOS, each time NAM is idled or aborted, the host's COLLECT utility writes certain NOS files to tape for long-term storage, including CDCNET dump files (COLLECT records the dump's CDCNET file name with each dump file collected).

COLLECT has an option to keep file types separate by writing them to different tapes. If this option is selected, your CDCNET dump files are written to tape as a set whenever COLLECT is invoked; otherwise, CDCNET dump files are interspersed with any other files collected. In either case, collected files are purged from the system.

See the NOS Version 2 Analysis Handbook for further information about COLLECT.

Using EDIT_CATALOG to Find Dump Files (NOS/VE Only)

Use the SCL command EDIT_CATALOG (abbreviated EDIC) to help locate available dump files. For example, you can use the following EDIC command to see a list of all systems for which a dump file subcatalog has been created:

```
EDIC $SYSTEM.CDCNET.DUMP
```

The screen display from this subcommand lists all systems for which there might be dump files cataloged. You can choose to VIEW the contents of any of these subcatalogs by using the EDIC function keys.

How To Use the Dump Analyzer Input File

NOS/VE Only

Under NOS/VE, if you specify an input file on the ANACD command, you are not prompted for subcommands. Instead, output is written to the active output file and control returns directly to NOS/VE when the Dump Analyzer reads a QUIT subcommand or reaches the end-of-information (EOI) on the input file.

By SCL convention, the Dump Analyzer terminates input file processing if it encounters any abnormal status of severity ERROR or greater. None of the subcommands after the one causing the error are processed. This restriction applies even if INPUT=$INPUT.

If the call to ANACD is embedded in an SCL procedure, the command stream is considered to be inside that procedure. In this case, INPUT=$INPUT is the only way to permit input from the terminal. Again, any error causes the Dump Analyzer to terminate.

NOS Only

Under NOS, if you specify an input file on the ANACD command (it must be local to your job), you are not prompted for subcommands. Instead, output is written to the active output file and control returns directly to NOS when the Dump Analyzer reads a QUIT subcommand, or reaches the end-of-information (EOI) on the input file.

See the section later in this chapter titled Sample Input File for NOS/VE or NOS for an example of useful input file subcommands.

How To Manage Dump Analyzer Output

NOS/VE Only

Under NOS/VE, output management is very flexible. You can easily alternate between displays sent directly to your terminal and displays written to other files. Moreover, you can edit output files without leaving the Dump Analyzer session you are engaged in.

If you do not specify an alternative output file on the ANACD command or on the Dump Analyzer subcommand being executed, displays from the Dump Analyzer are written to file $OUTPUT, which is connected to terminal output file OUTPUT. The SCL command CREATE_FILE_CONNECTION (CREFC) lets you direct $OUTPUT to other files as well.

If you do not specify an alternative output file on the ANACD command, but do so on the Dump Analyzer subcommand being executed, display from that subcommand is written to the specified file.

If you do specify an alternative output file on the ANACD command, output from the Dump Analyzer subcommands is directed to the specified file *except* for display from subcommands on which you specify a different output file. You may even specify file $OUTPUT for a subcommand output file. This provides you with an immediate display, when output would otherwise have been written directly to the output file named on the ANACD command.

These options are summarized in the following table:

Alternative output file on ANACD?	Active subcommand output file?	Output is written to:
NO	NO	$OUTPUT
YES	NO	ANACD output file
NO	YES	Subcommand output file
YES	YES	Subcommand output file (may be $OUTPUT)

Because NOS/VE lets you nest different utilities, you can edit a Dump Analyzer output file from within a Dump Analyzer session. Use the SCL command EDIT_FILE (EDIF) in response to the Dump Analyzer prompt, and specify the appropriate Dump Analyzer output file for its FILE parameter. When you finish editing the file by specifying QUIT, you are automatically returned to the Dump Analyzer session in progress.

This can be especially useful if you need to locate values in a long Dump Analyzer display (such as DISPLAY_MEMORY_MAP) before effectively continuing your Dump Analyzer session.

NOS Only

Under NOS, you can direct Dump Analyzer output using the O parameter on the ANACD command or the OUTPUT parameter on any subcommands that provide one.

If you do not specify an alternative output file on the ANACD command or on the Dump Analyzer subcommand being executed, displays from the Dump Analyzer are written to file OUTPUT, which is usually connected to the terminal.

If you do not specify an alternative output file on the ANACD command, but do so on the Dump Analyzer subcommand being executed, display from that subcommand is written to the specified file.

If you do specify an alternative output file on the ANACD command, output from the Dump Analyzer subcommands is directed to the specified file *except* for display from subcommands on which you specify a different output file. You can also specify OUTPUT for a subcommand output file, which provides you with an immediate display when output would otherwise have been written directly to the output file named on the ANACD command.

These options are summarized in the following table:

Alternative output file on ANACD?	Active subcommand output file?	Output is written to:
NO	NO	OUTPUT (usually attached to your terminal)
YES	NO	ANACD output file
NO	YES	Subcommand output file
YES	YES	Subcommand output file (may be OUTPUT)

You cannot edit a Dump Analyzer output file from within a Dump Analyzer session. If you need to look through an output file before continuing, QUIT the current Dump Analyzer session, edit the output file, and begin a new Dump Analyzer session using the ANACD command.

How To End a Dump Analyzer Session

Under both NOS/VE and NOS you can end a Dump Analyzer session and return control to NOS/VE or NOS with the Dump Analyzer QUIT subcommand, or with the user break 2 sequence entered at the terminal in response to the Dump Analyzer prompt.

```
QUIT
```

NOTE

If you enter the user break 2 sequence while a subcommand is being processed, only that subcommand is terminated and the Dump Analyzer prompt is issued.

How To Transfer Dump Files Between NOS/VE and NOS

To transfer a DI dump file from NOS to NOS/VE on a dual-state host, use the NOS/VE GET_FILE command and set the DATA_CONVERSION parameter to B64.

To transfer a DI dump file from NOS/VE to NOS on a dual-state host, use the NOS/VE REPLACE_FILE command and set the DATA_CONVERSION parameter to B64.

See the NOS/VE System Usage manual for further information about the GET_FILE and REPLACE_FILE commands.

Sample Input File for NOS/VE or NOS

By using a Dump Analyzer input file, a single ANACD command can be used to process a group of Dump Analyzer subcommands. Use the host's file editing utilities to create Dump Analyzer input files. Use the host's file management utilities to store and retrieve these files for routine use.

Following is a sample input file for use on NOS/VE or NOS. This set of subcommands reveals important information about the dump file and can help determine if further analysis is required.

```
DISPLAY_EXECUTIVE_ERROR_TABLE
DISPLAY_DI_SYSTEM_STATUS
DISPLAY_TASK_CONTROL_BLOCK TASK_IDENTIFIER=ALL
VALIDATE_GLOBAL_INFORMATION
DISPLAY_NETWORK_STATUS
DISPLAY_HARDWARE_STATUS DISPLAY_OPTION=FULL
DISPLAY_SYSTEM_CONFIGURATION_TABLE
DISPLAY_MEMORY ADDRESS=400 REPEAT_COUNT=320(10)
DISPLAY_MEMORY_MAP
DISPLAY_CALL TASK_IDENTIFIER=ALL
```

See the respective subcommand descriptions for explanations and examples of the information displayed from these subcommands.

Sample Output File for NOS/VE or NOS

Dump Analyzer output is written either to the screen of your terminal or to a specified output file.

If Dump Analyzer output is written to a separate output file, each page of output includes a header that identifies the level of Dump Analyzer you are using, date and time of subcommand execution, and the Dump Analyzer subcommand being executed. The subcommand is shown in its unabbreviated form, regardless of how it was entered. Parameters are echoed exactly as they were entered.

Following is a sample of output that was written to an output file:

```
1    COPYRIGHT CONTROL DATA CORPORATION 1985, 1986, 1987  ALL RIGHTS RESERVED
  ANACD - Level 4104     November 13, 1987     1:42 PM           PAGE        1
  DISPLAY_NETWORK_STATUS  o=netstat


 DISPLAY NETWORK STATUS

 network_name       = MTI_JONAS_NETWORK_1
 network_type       = HDLC
 network_identifier = 0000A004(16)
 network_status     = active
 network_cost       = 0037D(16)

 network_name       = MTI_JONAS_NETWORK_2
 network_type       = HDLC
 network_identifier = 0000A005(16)
 network_status     = active
 network_cost       = 0037D(16)
```

All Dump Analyzer output that is written to a separate output file uses this header convention.

NOTE

Subcommand lines up to 256 characters long are accepted and written to the display file without truncation, although printing on paper designed for 80 or 132 columns may result in loss of visible data.

Summary of ANACD Subcommands

ANACD subcommands processed by the Dump Analyzer fall into six major categories, as summarized in the following table.

Dump Analyzer Control Subcommand	Description
QUIT (QUI)	Terminates the Dump Analyzer session and returns control to the operating system.

Dump Analyzer Help Subcommands	Description
DISPLAY_COMMAND_INFORMATION (DISCI)	Displays parameter information for a specified Dump Analyzer subcommand.
DISPLAY_COMMAND_LIST (DISCL)	Displays a list of available Dump Analyzer subcommands.
HELP (HEL)	Displays a list of available Dump Analyzer subcommands, or parameter information for a specified Dump Analyzer subcommand.

Dump Summary Subcommands	Description
DISPLAY_AUTO_DUMP_TABLE (DISADT)	Displays the contents of the auto dump table and the map of memory available in the dump file.
DISPLAY_BOARD_MAP_TABLE (DISBMT)	Displays the contents of the board map table.
DISPLAY_DI_SYSTEM_STATUS (DISDSS)	Displays system configuration information from the time of reset.
DISPLAY_EXECUTIVE_ERROR_TABLE (DISEET)	Displays information from the executive error table.
DISPLAY_HARDWARE_STATUS (DISHS)	Displays information about the modular DI hardware from the time of reset.
DISPLAY_ICA_SYSTEM_STATUS (DISISS)	Displays system configuration information from the time of reset.
DISPLAY_NETWORK_STATUS (DISNS)	Displays status of networks connected to the DI at reset.
DISPLAY_SYSTEM_CONFIG_TABLE (DISSCT)	Displays information from the system configuration table.
VALIDATE_GLOBAL_INFORMATION (VALGI)	Displays general diagnostic information, serving as a preliminary guide to further analysis.

Address-Oriented Subcommands	Description
DISPLAY_BUFFER_CHAIN (DISBC)	Displays buffer chain information starting from the specified machine address. All descriptor buffers or data buffers are displayed for all messages in the chain.
DISPLAY_DATA_QUEUE (DISDQ)	Displays buffer chain information associated with the queue control block at the specified machine address.
DISPLAY_LINKED_LIST (DISLL)	Displays a linked list of structured elements given the machine address of the first element and offset to a pointer within the structure linking it to the next item in the list.
DISPLAY_MEMORY (DISM)	Displays memory contents. The display begins at a specified machine address or entry-point address and is of a specified length. Both hexadecimal and ASCII formats are displayed.
DISPLAY_MEMORY_HEADER (DISMH)	Displays information from the allocation header of the memory extent that contains the specified address.
DISPLAY_TREE (DIST)	Displays all nodes of a binary tree structure, or a specified node matching a user-supplied key.

Task-Oriented Subcommands	Description
DISPLAY_CALL (DISC)	Displays information about the dynamic call chain (or module traceback) of the specified task, or all tasks.
DISPLAY_MEMORY_USERS (DISMU)	Displays information about the allocation of memory extents to various tasks or other users in the system.
DISPLAY_TASK_CONTROL_BLOCK (DISTCB)	Displays information from the task control block for the specified task, or all tasks.
SELECT_TASK (SELT)	Specifies the task to be examined in subsequent task-oriented subcommands (designates the task to be considered the current task).
VALIDATE_STACK_AREAS (VALSA)	Checks integrity of the reserved and user stack areas of any or all tasks.

Miscellaneous Subcommands	Description
DISPLAY_LINE_CONTROL_BLOCK (DISLCB)	Displays information from the configured line control block for a given line, or all lines.
DISPLAY_LOG_QUEUES (DISLQ)	Displays the preserve and initialization log message queues.
DISPLAY_MEMORY_MAP (DISMM)	Displays a memory map of the modules that were loaded in the DI before it was reset.

Analyzing the Network 9

Analyzing the Network 9

Each network is a complex data communications facility with its own unique configuration of network solutions, DIs, hosts, terminals, and peripheral devices. Yet, despite the complexity and domain of CDCNET, it is designed so that the typical user is barely aware of its presence.

Trained network analysts must be able to discover the relationships among the varied CDCNET network components and help maintain the transparency of the network to the end user. The CDCNET product is equipped with tools to help the network analyst perform these functions.

Analysis Tools

The tools available for analyzing the network include the Network Operator Utility (NETOU), the DI Dump Analyzer, and the Network Performance Analyzer (NPA). Each of these tools reveals the network in a different way.

NETOU has a set of commands that let you look at the network from an active DI's perspective. Several of the NETOU commands cause a DI to display information that it maintains in its memory's data structures, or that it can find out from other active DIs on the network.

The DI Dump Analyzer can display some of the same data structures, but only from a host-based file that is created when a DI dumps its memory at reset. Instead of telling you about an active DI, the Dump Analyzer displays information about a specific point in time in the past—the point when the DI was forced to reload.

In contrast, NPA looks at the network over time. This tool generates statistics based on CDCNET log files. It can reveal trends and averages that are useful when analyzing the network's performance.

Table 9-1 summarizes each of these network analysis tools according to its mode of operation, its residence, and its specific application.

Table 9-1. Network Analysis Tools

Tool	Mode of Operation	Residence	Application
NETOU	Causes DIs to process commands	Host- and DI-based	Used to query active DIs
DI Dump Analyzer	Processes DI dump files	Host-based	Displays information from the time of DI reset
NPA	Processes CDCNET log files	Host-based	Reformats log files to provide statistical information from selected time periods or generates event/error reports based on log messages in the network log file

Among other things, these tools allow the network analyst to evaluate the following:

- Network configuration
- DI hardware configuration
- Lines and terminal connections

How Networks Are Formed

The structure of a network is a function of the network configuration commands entered interactively or via procedure files, and the physical status of network hardware components. By using the network analysis tools described in this chapter, you can discover and/or verify network configurations and topologies.

To help with your analysis, gather as much existing information as you can about how the network was formed. Examine the following:

- DI Configuration Procedures
- Exception List
- Terminal Definition Procedures
- Terminal User Procedures

Work with the site operations personnel and secure a map of the physical layout of the major network hardware components (hosts, DIs, and trunks), including the default and assigned names of these components. Determine if there is an online database describing the network and study it carefully if there is one. If you have been called in to analyze someone's network, begin by using your tools to verify the site's current base of information.

Analyzing Network Configuration

The roadmap of a concatenated network (or catenet) is a function of its major hardware configuration—the interconnection of its DIs, hosts, and trunks. Traffic patterns, bottlenecks, and detours are the result of configuration decisions and hardware conditions. See the CDCNET Configuration Guide for descriptions of the commands used to configure a CDCNET.

Figure 9-1 shows a sample network configuration with six DIs.

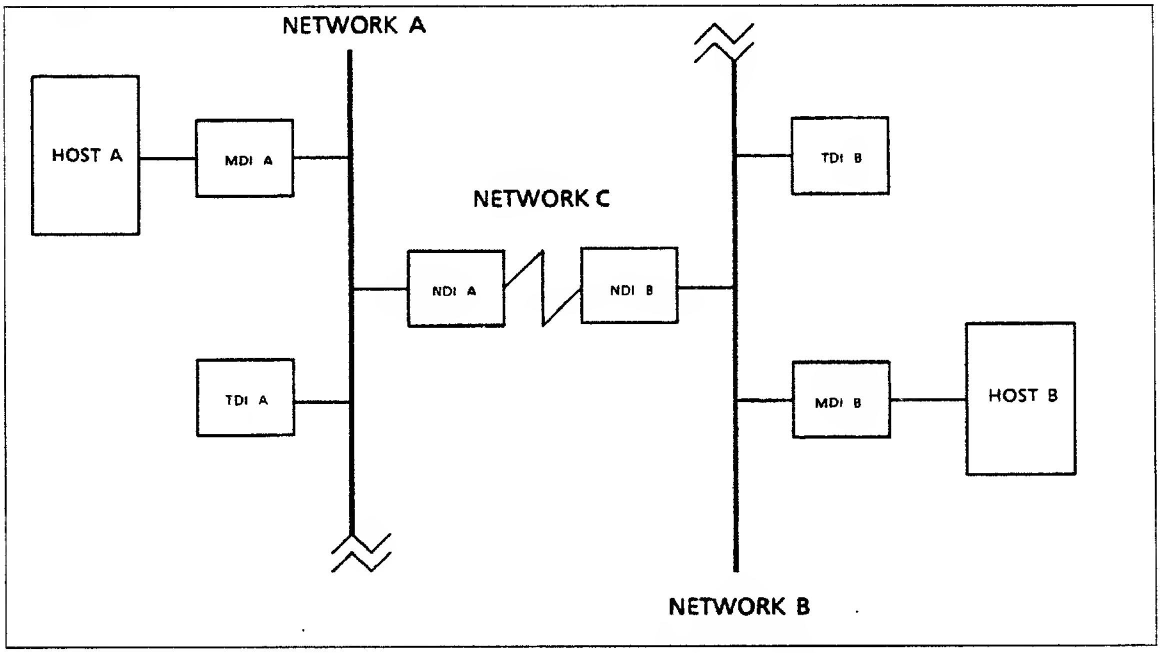


Figure 9-1. Network Configuration Map

Analyzing Network Configuration Using NETOU

After you have started a NETOU session, you can begin to query the individual CDCNET systems that service the catenet.

For example, assume that your network looks like the one in figure 9-1, that your terminal is connected to the network through TDI_A, and that you are unable to create a connection to HOST_B. You begin a NETOU session on (NOS) HOST_A, with MDI_A as the Command MDI, and find that you receive no response from commands sent to NDI_B, MDI_B, or TDI_B.

Looking at the map, you immediately suspect a problem in the link from network solution A to network solution B. You could check this hypothesis by sending a DISPLAY_NETWORK_STATUS (DISNS) command to NDI_A.

If you are plotting a configuration map from scratch, first discover the system titles (this is described in the next section, Displaying System Titles) and then use NETOU's DISPLAY_NETWORK_STATUS (DISNS) command (described later in this section). As you collect network hardware information, group the systems according to the network solutions they serve. Thus grouped, you should be able to draw a catenet configuration map like the one in figure 9-1. This map can help you probe the network and isolate potential trouble spots quickly.

Displaying System Titles

System titles are used in a NETOU session to identify CDCNET systems, as when you specify a name on the SENC command. If the title you specify cannot be located, the command you enter cannot be delivered for execution and an error results. It is therefore useful to know, at any given time, which DI systems are active on the catenet.

There are two system names for each CDCNET system that can be addressed: the default system name and the logical system name (if one was specified when the system was defined). Default names are of the form $DI_system_id, where system_id is the DI's 12-hexadecimal-digit system identifier.

Knowing the site conventions for assigning logical names can be very useful. A convention that makes an obvious connection between the alias and the default system name can save you time during analysis.

Because the default system name is based on a unique system identifier, tracking this name ensures that you do not confuse one CDCNET system for another. Keeping track of the system identifier is also useful if you plan to use NPA's CRECAR command. One of the parameters on the CRECAR command specifies the system identifier of the system to be covered in the report.

Displaying System Titles Under NOS

If you are using NETOU under NOS and want to display the active system titles, the Command MDI can make a system title search throughout the catenet. This search provides you with the titles of all CDCNET systems that the Command MDI can now address. You can order this search with the following command:

```
DISPLAY_CATENET_TITLES
```

or its abbreviated form:

```
DISCT
```

Display from DISCT is formatted as follows:

```
FROM MDI_8B                          33565
Catenet Titles

 system titles
  $DI_080025100083             TDI_81
  $DI_080025100091             $DI_080025100078
  MTI_83                       $DI_080025100083
  MDI_8B                       TDI_78
```

Displaying System Titles Under NOS/VE

There is no DISCT command under NOS/VE, but you can add an SCL procedure to your command library that has the same capability. The following procedure, called DISPLAY_SYSTEM_NAMES in the example, yields a display of active DI systems when executed as a command.

```
PROC display_system_names, dissn (
  system, s    string = '[A-Z[*'
  )

  x = $matching_names($translate(lower_to_upper, $value(system)))
  FOR i = $variable(x, lower_bound) TO $variable(x, upper_bound) DO
    disv x(i)
  FOREND

PROCEND display_system_names
```

If you need help adding this procedure to your command library, see the NOS/VE System Usage manual. If you have successfully added this procedure, simply type DISSN from within a NETOU session under NOS/VE to see a display of the logical names of all DI systems that are active on the network. For a display of the logical *and* default system names, type:

```
DISSN '*'
```

The asterisk is a wildcard character on NETOU. When used in this way, with this command, it enables matching of system names that begin with any character (not just alphabetic), and hence the default system names, which begin with a dollar sign ($), are included in the display.

The asterisk wildcard character can be used with this command in other ways, too. For example, to display only the default system names, type:

```
DISSN '$*'
```

DISPLAY_DI_SYSTEM_STATUS Command

Use the DISPLAY_DI_SYSTEM_STATUS (DISDSS) command to correlate a system's default name and its site-given alias. Simply send a DISDSS command to one of the titles in question.

For example, the following command displays the system name(s) of the DI whose default system name is $DI_080025300091:

```
SENC 'DISDSS' $DI_080025300091
```

Output from this command is formatted as follows:

```
FROM TDI_91                         33115

DI System Status
system name = TDI_91
system address = 080025300091(16)
boot version number = 190B(16)
software release level = 190B(16)
number of tasks = 54
free SMM memory = 64526
percent CPU utilization = 7
buffer state = good
memory state = good
date and time of last reload = 86/10/09  06.58.30

Buffer Status
type         total buffers  available buffers  buffer size
data         1225           886                144
descriptor   410            362                32

SMM Memory Status
total memory  available memory  fragments  deloadable memory
1048576       64526             96         83938

MPB RAM Status
total memory  available memory  fragments  deloadable memory
16384         1820              1          0
```

The site-assigned logical name of this DI is listed next to the heading system name. The default system name is constructed by prefixing the system address with the four-character string $DI_. If you know one name and not the other, you can always find the other by sending DISDSS to the name you know.

DISPLAY_NETWORK_STATUS Command

The network solutions served by a CDCNET system can be displayed by sending the DISPLAY_NETWORK_STATUS (DISNS) command to the system of interest. For each network solution directly connected to the CDCNET system, the network solution name, type, id, status, and cost are displayed. The name of the CDCNET system hardware device that services each of the network solutions (such as ESCI or LIM/PORT) is also displayed.

For example, the following command displays the network solutions directly connected to CDCNET system AHL_TDI_30011c:

```
SENC 'DISNS' AHL_TDI_30011C
```

Output from this command is formatted as follows:

```
FROM AHL_TDI_30011C

 Network Status
 network_name = LCLSH_ETHER_NET
 network_type = Ethernet
 network_id = 0000A002(16)
 network_status = active
 network_cost = 000A(16)
 trunk_name = LCLSH_ETHER_NET
 device_name = $ESCI7
 average time network is congested = 0 %
 date and time network last became active = 89/10/02  05.17 15


 network_name = RTI_LINK_NETWORK_1
 network_type = HDLC
 network_id = 0000A205(16)
 network_status = active
 network_cost = 06FA(16)
 trunk_name = RTI_LINK_NETWORK_1
 device_name = $LIM7_PORT0
 average time network is congested = ,0 %
 date and time network last became active = 89/10/02  05.17 45
```

Like CDCNET systems, network solutions can be known by more than one name. One system on the catenet might refer to a network solution by one name, while another system refers to the same network solution by a different name.

The common denominator for referencing network solutions is the network_id. Use the network_id to gather information about network solutions on a catenet. This number is the same for all references to the same network solution and is unique on the catenet.

For example, the network_id for the first of the two networks in the previous display is 0000A002(16), and the network_name is LCLSH_ETHER_NET. Any other system must refer to this network solution by the same network_id, but might use a different network_name.

Analyzing Network Configuration Using the Dump Analyzer

The Dump Analyzer is equipped with its own DISPLAY_DI_SYSTEM_STATUS and DISPLAY_NETWORK_STATUS subcommands. These subcommands reveal information that is similar to what their NETOU counterparts display, but from a DI dump file instead of from an active DI.

DISPLAY_DI_SYSTEM_STATUS Subcommand

The Dump Analyzer's DISPLAY_DI_SYSTEM_STATUS (DISDSS) subcommand is designed to reveal much of the same information as the NETOU command that has the same name. During a DI Dump Analyzer session, simply enter:

```
DISDSS
```

Output from this subcommand is formatted as follows:

```
DISPLAY DI SYSTEM STATUS

 System name       = MDI_84
 System identifier = 080025100084(16)     Master clock    =        FALSE
 Release level     =         2007         Number of tasks =           31
 Boot version      =         DDDD         CPU utilization =            5 %
 DI loaded from  MCI board in slot 7      Helping system  = 000000000000(16)

 Buffers
              free        total         size
 Data           81          451          144         State is GOOD
 Desc           39          100           32

 Memory      (1 MB Configuration)
              free    fragments    deloadable
 MPB          3074            1             0
 PMM         49258            2             0        State is GOOD
 SMM        346390            6         83394
 RESERVED     1000            1           N/A

 50% of memory after configuration made into buffers
 --WARNING DA 118--  Address  980001(16) is not within valid memory_
 ranges.

 Largest SMM memory fragment available =    65537
```

DISPLAY_NETWORK_STATUS Subcommand

Again, the DISPLAY_NETWORK_STATUS (DISNS) subcommand is designed to reveal much of the same information as the NETOU command by the same name. During a DI Dump Analyzer session, simply enter:

```
DISNS
```

Output from this subcommand is formatted as follows:

```
DISPLAY NETWORK STATUS

network_name       = $NET_1
network_type       = Ethernet
network_identifier = 00000001(16)
network_status     = active
network_cost       = 0000A(16)
```

This example shows the DISNS display for a DI that has one directly connected network. Following is an example of DISNS output for a DI that had no directly connected networks:

```
DISPLAY NETWORK STATUS
No networks defined
```

Analyzing DI Hardware Configurations

The modularity of DI hardware design permits dozens of DI hardware configurations. Standard, generalized configuration types are the TDI, the MDI, the MTI, the NDI, and the RTI. Each of these DI types can have further variation, so that the DI can be matched to an exact and specialized role in the network. For example, some DIs have two SMM boards; others have two CIMs.

Every DI has an MPB and at least one SMM board. Following are some sample DI hardware configurations (board slot numbers follow board type abbreviations; for example, MCI6 is an MCI board in slot 6):

DI_A	DI_B	DI_C	DI_D	DI_E
MPB0	MPB0	MPB0	MPB0	MPB0
SMM1	PMM1	PMM1	SMM2	PMM1
SMM2	SMM2	SMM2	CIM4	SMM2
CIM5	SMM3	CIM5	ESCI6	SMM3
ESCI6	CIM4	ESCI7		ESCI5
	CIM5			MCI7
	MCI6			
	ESCI7			

It is a good idea to keep up-to-date records of DI hardware configurations. This information can make network analysis much easier.

The DI maintains its own hardware status tables in order to record the status of its internal hardware configuration. Software components use these tables to update and access information about the modular hardware installed in a DI.

There are five types of DI hardware status tables. These types, and the functions they perform, are listed in table 9-2:

Table 9-2. Summary of Board Status Tables

Hardware Status Table	Description
Major Card Status Table (MCST)	This table has one entry for each major board installed in the DI. Each entry records the identity, status, and location of a major board. For a CIM, SMM, or PMM board, the entry also records a pointer to the next appropriate hardware status table.
LIM Status Table (LST)	This table has one entry for each LIM or URI installed in the DI. Each entry records the identity, status, and location of a LIM/URI. The entry also records the major card slot number of the parent CIM board and the address of a port status table for the ports serviced by the LIM/URI.
Port Status Table (PST)	Each LST entry may have an associated PST. In a PST there is one entry for each port serviced by the controlling LIM/URI. Each PST entry records the identity, status, and location of a port.
SMM Bank Status Table (SBST)	This table records information about the memory banks on the SMM board. It records the size and location of each SMM memory bank and the status of the SMM board itself.
PMM Bank Status Table (PBST)	This table records information about the memory banks on an installed PMM board. It records information about the PMM memory bank and the status of the PMM board itself.

Figure 9-2 illustrates the relationships among the various hardware status tables maintained in DI memory. The configuration shown in the MCST is arbitrary. Only the MPB board must reside in the designated position, slot 0 (a PMM, if present, must reside in slot 1).

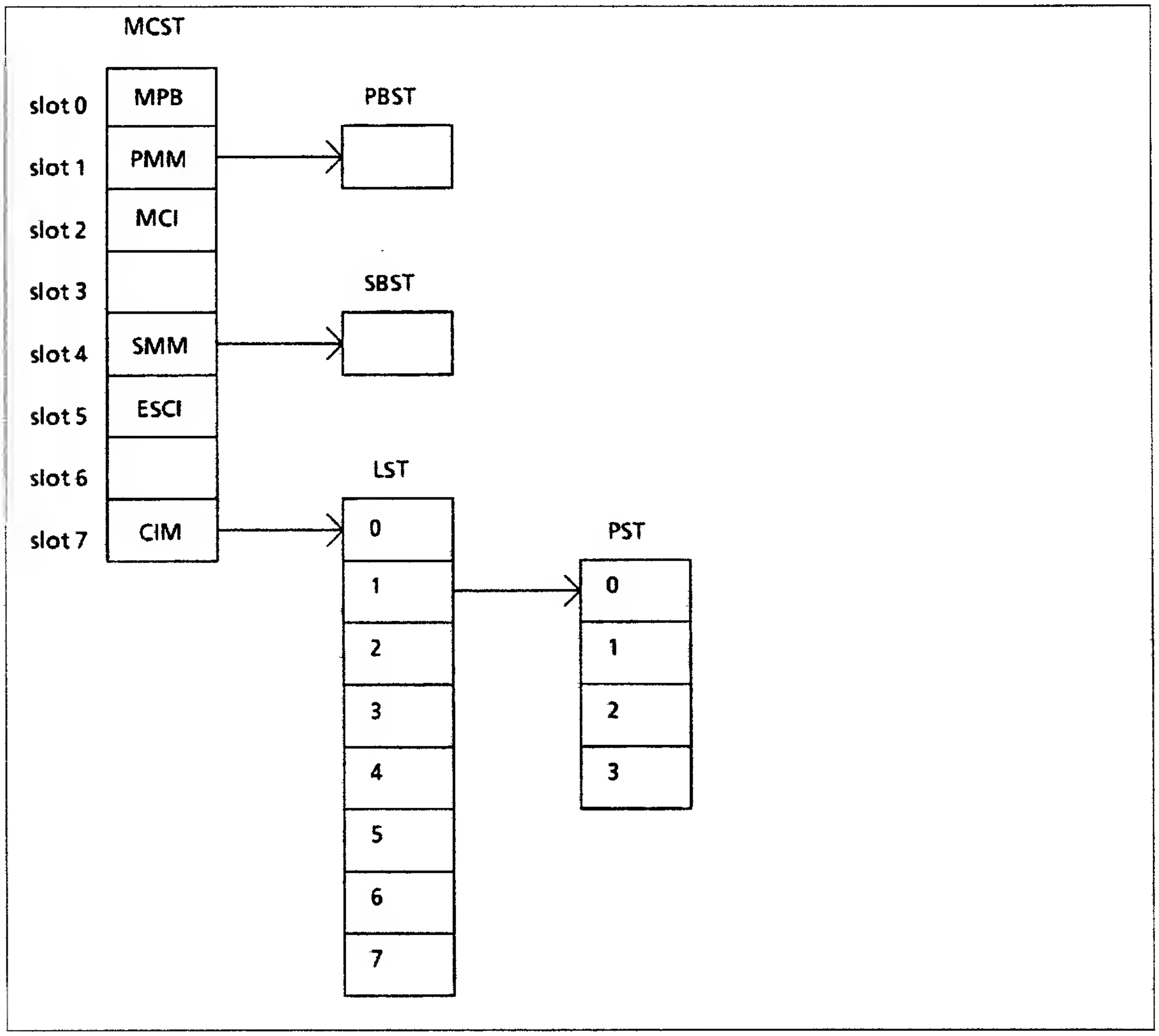


Figure 9-2. Summary of Board Status Table Relationships

The hardware status tables indicate the state and status of hardware devices internal to the DI. The range of values for state and status types is described in tables 9-3 and 9-4, respectively. The hexadecimal digit values in these tables can be used to interpret certain memory locations described in this chapter.

Table 9-3. Device State

State	Hex Digits	Description
On	0000	This state implies that the device is either fully operational or is operating in a degraded mode. This state further implies that the device is available for use by the system, which may include concurrent diagnostic maintenance.
Off	0001	This state indicates that the device is not available to the system or diagnostics. A device may be placed in this state to prohibit its further use. The logical equivalent of this state is that the device is not present in the DI.
Down	0002	This state indicates that the device is only available for use by the diagnostics software. The device is not available for normal system use.

Table 9-4. Device Status

Status	Hex Digits	Description
Not Configured	0000	The hardware device has not been configured.
Configured	0001	The hardware device has been configured, but not enabled.
Enabled	0002	The hardware device has been configured and the software driving the device has been started; however, the device is not currently active.
Active	0003	The hardware device is active.
Protocol Mismatch	0004	The hardware device has been configured and the software driving the device has been started; however, the software has detected a protocol mismatch with its peer. (This state is currently used only for the MCI board, to indicate that the MCI driver could not negotiate a common channel protocol with a NOS host.)

The hardware configuration and status of active DIs can be discovered using NETOU. Use the Dump Analyzer to find the same information about a DI that has dumped its memory. NPA can generate reports about hardware configuration and status over time.

Analyzing DI Hardware Configurations Using NETOU

NETOU is equipped with a command called DISPLAY_HARDWARE_STATUS (DISHS) that causes the DI to display information from its hardware status tables. In the following example, the hardware status of an MDI is displayed (note that DISHS must be enclosed in a SENC command):

```
SENC 'DISHS' MDI_A1
```

Output from this command is formatted as follows:

```
FROM MDI_A1                        33021

Hardware Status
device name  state   status        version     lim/bank/port  type
$MPB0          on    configured    0000(16)
$PMM1          on    configured    0008(16)
$SMM2          on    configured    0002(16)
3              off   not config.   0000(16)
4              off   not config.   0000(16)
$ESCI5         on    active        0010(16)
6              off   not config.   0000(16)
$MCI7          on    active        0000(16)
```

Following is an example of the display format for a TDI.

```
FROM TDI_A1                        33021

Hardware Status
device name  state   status        version     lim/bank/port  type
$MPB0          on    configureo    0000(16)
1              off   not config.   0000(16)
$SMM2          on    configured    0002(16)
$CIM3          on    configureo    0000(16)    0,1,2,3,4,5,6,7
4              off   not config.   0000(16)
$ESCI5         on    active        0000(16)
6              off   not config.   0000(16)
7              off   not config.   0000(16)
$LIM0          on    configured                4              RS232
$LIM1          on    configured                4              RS232
$LIM2          on    configured                4              RS232
$LIM3          on    configured                4              RS232
$LIM4          on    configured                4              RS232
$LIM5          on    configured                4              RS232
$LIM6          on    configured                4              RS232
$LIM7          on    configured                4              RS232
```

Analyzing DI Hardware Configurations Using the Dump Analyzer

With the Dump Analyzer, you can look at DI hardware status either by using a specialized subcommand or by looking at the memory corresponding to the DI's hardware status tables.

DISPLAY_HARDWARE_STATUS Subcommand

The Dump Analyzer is equipped with a subcommand that displays information in the MCST and LST from the dump file. This subcommand, called DISPLAY_HARDWARE_STATUS (DISHS), is very much like its NETOU counterpart. During a DI Dump Analyzer session, simply enter:

```
DISHS
```

Output from this subcommand is formatted as follows:

```
DI HARDWARE STATUS

      Card  Card    Boot                          Version    ROM      Dump
Slot  type    ok  enabled  State       Status       (16)    level  address (16)
--------------------------------------------------------------------------------
  0    MPB   yes       no     on   configured          0     50C          0
  1  EMPTY
  2    SMM   yes       no     on   configured          2       0     100000
  3    SMM   yes       no     on   configured          0       0     200000
  4  EMPTY
  5    CIM   yes       no     on   configured          0     50C      90000
  6    CIM   yes       no     on   configured          0     50C      A0000
  7   ESCI   yes      yes     on       active         10     806      B0000

Line Interface Modules

Slot            State      Status   Parent CIM  LIM Type
---------------------------------------------------------
  0                on  configured       $CIM6  rs232
  1                on  configured       $CIM6  rs232
  2                on  configured       $CIM6  rs232
  3                on  configured       $CIM6  rs232
  4                on  configured       $CIM6  rs232
  5                on  configured       $CIM6  rs232
  6                on  configured       $CIM6  rs232
  7                on  configured       $CIM5  rs232
```

If you need to look at the memory corresponding to the individual DI hardware status tables, you can use the Dump analyzer's DISPLAY_MEMORY (DISM) subcommand, as described in the following subsections.

Locating and Interpreting the Major Card Status Table (MCST)

The MCST is the root through which all other hardware status tables in DI memory can be located. To first locate the MCST, use the Dump Analyzer DISPLAY_MEMORY_MAP subcommand, or address the MCST symbolically as major_card_status_table. The MCST contains 52 bytes of information for each board slot. Its total length is 416 bytes.

Because each entry in the MCST has the same length, it is easy to display the MCST entry by entry. To do this, use the following subcommand:

```
DISM A=MAJOR_CARD_STATUS_TABLE BC=52(10) RC=8
```

An example of the display from this subcommand appears below. Each entry begins on a new line, making it easy to locate the same field in more than one entry.

```
STARTING ADDRESS     151280

HEX AODR              HEXADECIMAL OATA                     ASCII DATA

 151280  2440 5042 3020 2020 2020 2000 0000 0001   $MPB0
         0000 0000 0000 050C 0000 0000 0000 0000
         0000 0000 0000 0000 0000 0000 0000 0000
         0010 32BA                                   2
 1512B4  2450 4040 3120 2020 2020 2000 0000 0001   $PMM1
         0001 0008 0008 050C 0000 8000 0000 0000
         0000 0000 0000 0000 0000 0000 0010 3352                3R
         0010 3306                                   3
 1512E8  2453 4040 3220 2020 2020 2000 0000 0001   $SMM2
         0002 0009 0002 0000 FFFC 8800 0000 00FB
         0090 0000 0000 6011 0010 0000 0010 33C4               3
         0010 3378                                   3x
 15131C  2020 2020 2020 2020 2020 2000 0001 0000
         0003 000F 0000 0000 FFFF 0000 0000 0000
         0000 0000 0000 0000 0000 0000 0000 0000
         0000 0000
 151350  2443 494D 3420 2020 2020 2000 0000 0001   $CIM4
         0004 0001 0000 050C 0000 8880 FF00 00F7
         00A0 0000 0000 6021 0008 0000 0015 1420       `
         0010 3436                                   46
 151384  2020 2020 2020 2020 2020 2000 0001 0000
         0005 000F 0000 0000 FFFF 0000 0000 0000
         0000 0000 0000 0000 0000 0000 0000 0000
         0000 0000
 1513B8  2445 5343 4936 2020 2020 2000 0000 0003   $ESCI6
         0006 0002 0010 0806 FFFE 8800 0000 00F3
         00B0 0000 0000 6031 000A 0000 001B 365A       `1      6Z
         0010 4522                                   E"
 1513EC  244D 4349 3720 2020 2020 2000 0000 0003   $MCI7
         0007 0000 0000 050C FFFE C000 0000 00B6
         00B8 0000 0000 6039 000B 0000 0019 752E       `9      u
         0010 456E                                   En
```

Each 52-byte-long segment from the MCST is an entry that records information about a single DI board. Because each entry has the same structure, the following discussion applies equally to any entry in the MCST.

Board Name

The name of each major board is recorded in bytes 0 through 10 of its MCST entry. Each name begins with a dollar sign (hexadecimal 24), indicates the board type, and is completed with the board slot number. For example, $MCI7. The remaining length of the name field is blank-filled (hexadecimal 20).

Board State

The board state is recorded in bytes 12 and 13 of the MCST entry. See table 9-3 for explanations of the range of values for these bytes. In the previous DISPLAY_ MEMORY example, the MCST entries for slots 3 and 5 show the device state is off. No boards were installed in those slots when the DI was reset.

Board Status

The board status is recorded in bytes 14 and 15 of the MCST entry. See table 9-4 for explanations of these values. In the previous DISPLAY_MEMORY example, the MCST entries for slots 3 and 5 show the device status is not_configured. No boards were installed in those slots when the DI was reset.

There are five boolean values coded into byte 26 that further record the board's status. Following is a description of these boolean values.

Bit	Significance
0	If 1, the board passed diagnostics (not applicable to MPB)
1	If 1, the board is available
2	If 1, the board is running in degraded mode
3	If 1, the attention switch on the board has been set
4	If 1, the DI may not be booted over this board (not applicable to ESCI, CIM, or MCI)

Board Version Number and ROM Level

The version number of the installed board is recorded in bytes 20 and 21. Bytes 22 and 23 record the board's ROM level.

Bus Registers and Addresses

The value of Internal Control Bus (ICB) write register 0 is recorded in bytes 30 and 31 of each MCST entry. The ICB write register 1 value is in bytes 32 and 33.

The ICB address is in bytes 36 through 39. Bytes 40 through 43 record the ITB address.

Board Type-Specific Information

Information specific to the MPB, CIM, or ESCI board is stored in byte 27. The significance of this byte is as follows:

Board Type	Significance
MPB	Number of errors since last reset
CIM	Number of LIMs supported
ESCI	A boolean value that indicates whether the ESCI transceiver is bad; if byte 28 is 80(16) or greater, the transceiver is bad

There is an address in bytes 44 through 47 that points to further information for certain board types, as follows:

Board Type	Further Information
PMM	PMM Bank Status Table (PBST)
MCI	Link Interface Block (LIB)
SMM	SMM Bank Status Table (SBST)
CIM	LIM Status Table (LST)

Locating and Interpreting the LIM Status Table (LST)

There is a single LST in DI memory that records information about all LIM boards in the DI. It maintains a 38-byte-long entry for each LIM slot.

There are three ways to find the LST:

1. Bytes 44 through 47 of a CIM board's MCST entry point to the first entry in the LST that belongs to a LIM it controls.
2. The Dump Analyzer DISPLAY_MEMORY_MAP subcommand displays an address for the lim_status_table.
3. The LST can be addressed symbolically as lim_status_table.

To display the entire LST, use the following subcommand (readability of the LST display is improved if you specify a byte count the length of a single LST entry and a repeat count equal to the total number of entries):

```
DISM A=LIM_STATUS_TABLE BC=38(10) RC=8
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    151420

HEX ADDR              HEXADECIMAL DATA                     ASCII DATA

  151420   244C 494D 3020 2020 2020 2000 0000 0001    $LIM0
           0010 34CE 0000 0100 0000 0001 000F 0004      4
           0004 0010 3482                                 4
  151446   244C 4940 3120 2020 2020 2000 0000 0001    $LIM1
           0010 36E2 0001 0100 0000 0001 000F 0004      6
           0004 0010 3696                                 6
  15146C   244C 494D 3220 2020 2020 2000 0000 0001    $LIM2
           0010 38F6 0002 0100 0000 0001 000F 0004      8
           0004 0010 38AA                                 8
  151492   244C 494D 3320 2020 2020 2000 0000 0001    $LIM3
           0010 3B0A 0003 0100 0000 0001 000F 0004      ;
           0004 0010 3ABE                                 :
  1514B8   244C 494D 3420 2020 2020 2000 0000 0001    $LIM4
           0010 3D1E 0004 0100 0000 0001 000F 0004      =
           0004 0010 3CD2                                 <
  1514DE   244C 494D 3520 2020 2020 2000 0000 0001    $LIM5
           0010 3F32 0005 0100 0000 0001 000F 0004      ?2
           0004 0010 3EE6                                 >
  151504   244C 494D 3620 2020 2020 2000 0000 0001    $LIM6
           0010 4146 0006 0100 0000 0001 000F 0004      AF
           0004 0010 40FA                                 @
  15152A   244C 494D 3720 2020 2020 2000 0000 0001    $LIM7
           0010 435A 0007 0100 0000 0001 000F 0004      CZ
           0004 0010 430E                                 C
```

Each 38-byte entry in the LST records information about a single LIM board. Because each entry has the same structure, the following discussion applies equally to any entry in the LST.

LIM Name

The name of each LIM board is recorded in bytes 0 through 10 of its LST entry. Each name is of the form $LIMn, where n is the LIM slot number. For example, $LIM7. The remaining bytes of the name field are blank-filled (hexadecimal 20).

LIM State

The state of each LIM board is recorded in bytes 12 and 13 of its LST entry. See table 9-3 for explanations of the values for these bytes. In the example, all LIMs are on. A value of 01 in byte 23 indicates that LIM service is degraded.

LIM Status

LIM status is recorded in bytes 14 and 15 of each LST entry. See table 9-4 for explanations of the values for these bytes. In the example, all LIMs are configured.

LIM Type

The LIM type is recorded in bytes 26 and 27 of each LST entry. The hexadecimal value of these bytes has the following significance:

Hex Digits	Significance
0000	RS-449
0001	RS-232
0002	URD_BP1500
0003	URD_B300
0004	URD_E_SERIES
0005	URD_LINE_WRITER
0006	URD_FASTBAND
0007	URD_DATA_PRODUCTS_BASICS
0008	URD_CENTRONICS_360X_720X
0009	URD_CENTRONICS_703
000A	V35
000B	X21
000C	LIM_SLOT_EMPTY

All LIMs in the example are type RS-232.

The Owning CIM, Owned Ports

The major board slot number of the CIM board that is parent to a LIM is recorded in bytes 30 and 31 of the LIM's LST entry. Bytes 32 and 33 indicate the number of ports that the LIM owns; this number ranges from 0 through 0F(16). Each LIM board in the example owns four ports.

Locating and Interpreting a Port Status Table (PST)

The PST is a data structure that maintains information about the ports controlled by a single LIM. It has one entry for each port serviced by the LIM (currently there is a maximum of eight ports per LIM).

Bytes 16 through 19 of the LST entry for the controlling LIM board point to its PST. Use this address with the Dump Analyzer DISPLAY_MEMORY subcommand to display 28 bytes for each port.

To display the entire PST associated with LIM 0 of the previous example, use the following subcommand (readability of this display is improved if you specify a byte count the length of a single PST entry and a repeat count equal to the total number of entries in the PST):

```
DISM A=1034CE BC=38(10) RC=4
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS     1034CE

HEX ADDR             HEXADECIMAL DATA                    ASCII DATA

  1034CE  244C 494D 305F 504F 5254 30D4 0000 0002    $LIM0_PORT0
          0000 0003 001C 10DA 0010 3566                         5f
  1034EA  244C 494D 305F 504F 5254 311A 0000 0002    $LIM0_PORT1
          0001 0003 001E BE56 0010 35B2                     V  5
  103506  244C 494D 305F 504F 5254 324B 0000 0002    $LIM0_PORT2K
          0002 0003 001E C5E8 0010 35FE                        5
  103522  244C 494D 305F 504F 5254 33EC 0000 0002    $LIM0_PORT3
          0003 0003 001E C52A 0010 364A                      *  6J
```

Each 28-byte-long entry from the PST records information about a single port. Because each entry has the same structure, the following discussion applies equally to any entry in the PST.

Port Name

The name of each port is recorded in bytes 0 through 10 of its PST entry. Each port name is of the form $LIMn_PORTm, where n is the slot number of the parent LIM and m is the number of the port associated with the PST entry. The name field is blank-filled (hexadecimal 20).

Port State

The state of each port is recorded in bytes 12 and 13 of its PST entry. See table 9-3 for explanations of state values. All ports in the previous example are on.

Port Status

The status of each port is recorded in bytes 14 and 15 of its PST entry. See table 9-4. All ports in the previous example are enabled.

Port User

The port user type is recorded in bytes 18 and 19 of each PST entry. These bytes have the following significance:

Hex Digits	Significance
0000	Port available
0001	HDLC owner
0002	X.25 owner
0003	LCM owner

All ports in the previous example are owned by the Line Control Module (LCM).

Bytes 20 through 23 of the PST point to the port user configuration table. The type of table depends on the port owner type, as follows:

Port Owner	Table Type
HDLC	HDLC Link Interface Block (LIB)
X.25	X.25 LIB
LCM	Configured Line Control Block (CLCB); see the section of this chapter on Analyzing Line and Terminal Connections

Locating and Interpreting an SMM Bank Status Table (SBST)

The SBST is a data structure that records the size, location, and status of a DI's SMM memory banks. There is a single SBST assigned for each SMM board installed in the DI. An SBST is 54 bytes long.

There is a pointer to the SBST in bytes 44 through 47 of the SMM board's MCST entry. The SBST for the SMM board in the MCST example in this chapter is displayed with the following subcommand:

```
DISM A=1033C4 RC=54(10)
```

Output from this subcommand is formatted as follows.

```
STARTING ADDRESS     1033C4

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

  1033C4  0002 0008 0000 2453 4D4D 325F 4241 4E4B          $SMM2_BANK
  1033D4  3020 0000 0001 0010 0000 0000 0000 2453     0              $S
  1033E4  4D4D 325F 4241 4E4B 31EF 0000 0001 0018     MM2_BANK1
  1033F4  0000 0000 0000
```

SMM Memory Banks

Each SMM board has either one or two memory banks. Bytes 0 and 1 of the SBST indicate the number of memory banks on the associated SMM board. Bytes 2 through 5 represent the size of the SMM memory banks in bytes. In the example, there are two memory banks, each of size 80000(16), or 1/2 megabyte.

The name of the first memory bank (bank 0) is recorded in bytes 6 through 16 of the SBST entry. This name is of the form $SMMn_BANK0, where n is the board slot number of the SMM board. The remaining length of the name field is blank-filled (hexadecimal 20). The name of the second memory bank (bank 1) is recorded in bytes 30 through 40.

The codes for the state and status of each memory bank, and its starting address, are found in the following byte locations of the SBST:

	State	Status	Starting Address
BANK0	Bytes 18-19	Bytes 20-21	Bytes 22-25
BANK1	Bytes 42-43	Bytes 44-45	Bytes 46-49

Locating and Interpreting a PMM Bank Status Table (PBST)

The PBST is a data structure that records the size, location, and status of a DI's PMM memory. There is a single PBST created for each PMM board installed in the DI. Each PBST is 24 bytes long.

There is a pointer to the PBST in bytes 44 through 47 of the PMM board's MCST entry. The PBST for the PMM board in the MCST example in this chapter is displayed with the following subcommand:

```
DISM A=103352 RC=24(10)
```

Output from this subcommand is formatted as follows.

```
STARTING ADDRESS:   103352

HEX ADDR            HEXADECIMAL DATA                    ASCII DATA

  103352  0002 0000 2450 4D4D 315F 4241 4E4B 309E        $PMM1_BANK0
  103362  0000 0001 0000 0000
```

Each PMM has a single memory bank. Bytes 0 through 3 of the PBST indicate the size of that memory bank in bytes. The PMM board in the example has 128 K bytes of memory.

The name of the PMM memory bank is recorded in bytes 4 through 14 of the PBST entry. This name is of the form $PMMn_BANK0, where n is the board slot of the PMM board. The remaining length of the name field is blank-filled (hexadecimal 20).

Bytes 16 and 17 record the state of the PMM memory bank. See table 9-3.

Bytes 18 and 19 record the status of the PMM memory bank. See table 9-4.

Analyzing DI Hardware Configurations Using NPA

NPA's CRECAR command reports on the DI hardware *and* software. This report is called CONFRP1. An example of CONFRP1 appears below (the actual report appears on two pages):

```
1
                                                              88/07/26



                          NETWORK PERFORMANCE ANALYZER
                               VERSION 1.10/5303

                          CDCNET CONFIGURATION MESSAGES
                          SORTED BY DI, DATE, AND TIME

                                 CONFRP1 REPORT

                      TIME PERIOD = 00/01/01 0000 - 99/12/31 2400

                            SYSTEM IDS SELECTED = ALL


                            LOG IDS SELECTED    = ALL


                            LOG IDS EXCLUDED    = NONE


1
REPORT DAY  88/07/23                                         PAGE     1
                                CONFRP1 REPORT
          TITLE = AHD_NDI_10006D                      SID = 08002510006D

  DATE      TIME       SYSTEM ID    LOG ID
======== =========== ============  ======

88/07/23 00 55 00230 08002510006D     593
DI SYSTEM STATUS
SYSTEM NAME = AHD_NDI_10006D
SYSTEM ADDRESS = 08002510006D(16)
BOOT VERSION NUMBER = 5303(16)
SOFTWARE RELEASE LEVEL = 5303(16)
NUMBER OF TASKS = 23
FREE SMM MEMORY = 244946
PERCENT CPU UTILIZATION = 31
BUFFER STATE = GOOD
MEMORY STATE = GOOD
DATE AND TIME OF LAST RELOAD = 88/07/09  09.01.52
-
BUFFER STATUS
TYPE          TOTAL BUFFERS  AVAILABLE BUFFERS  BUFFER SIZE
DATA          1752           1091               144
DESCRIPTOR    571            547                32
-
SMM MEMORY STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
1048576       244946            123        57146
-
PMM MEMORY STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
131072        4912              1          0
-
MPB RAM STATUS
TOTAL MEMORY  AVAILABLE MEMORY  EXTENTS    DELOADABLE MEMORY
16384         1794              1          0
LARGEST SMM MEMORY EXTENT AVAILABLE = 218840
```

Analyzing Line and Terminal Connections

CDCNET terminal users reach the network over communications lines or connections. Being able to analyze these connections is highly important for the following reason: CDCNET operators and analysts are in the business of providing a reliable service to terminal users.

The DI maintains a series of chained data structures to record information about terminals connected to a TDI or MTI. The end user's connection can be traced through these data structures with the help of network analysis tools.

The five types of line and terminal control blocks are listed in table 9-5. They are listed in order, from the most aggregated to the least aggregated (from the control block root for the whole DI to the control block for an individual data connection).

Table 9-5. Summary of Terminal Control Blocks

Terminal Control Block	Description
Allocated Line Control Block (ALCB)	The ALCB is the terminal control block root. It points to the first element in the CLCB chain (described next). There is one ALCB, and it remains in DI memory whether or not any lines are actually configured.
Configured Line Control Block (CLCB)	There is one CLCB for each one of the DI's configured lines. Each CLCB records information about the line attributes, the locations of its owning ALCB, the next element in the CLCB chain (if any), and the first element in the TCCB chain it controls (described next). This control block is built by the Line Control Module (LCM) whenever a new line is defined.
Terminal Cluster Control Block (TCCB)	There is one TCCB for each set of clustered terminal devices (including sets of one terminal). Each TCCB maintains terminal accounting information and records the locations of its owning CLCB, the next element in the TCCB chain (if any), and the first element in the TDCB chain it controls (described next). There can be as many TCCBs as the number and capacities of configured lines in the DI permit.

(Continued)

Table 9-5. Summary of Terminal Control Blocks *(Continued)*

Terminal Control Block	Description
Terminal Device Control Block (TDCB)	There is one TDCB for each configured terminal device. Each TDCB records the associated terminal's device status, definition, accounting, and protocol information. It also records the locations of its owning TCCB, the next element in the TDCB chain (if any), and the first element in the DCCB chain it controls (described next). The LCM builds a TDCB when a new terminal or batch device is defined. If the new terminal device is the first one for a terminal cluster, the LCM also constructs a TCCB.
Data Connection Control Block (DCCB)	There is one DCCB for each data connection to a configured terminal device, whether unidirectional or bidirectional. Each DCCB records connection parameters and attributes, TDSM-defined values, the locations of its owning TDCB, the next element in the DCCB chain (if any), and an output buffer queue. Site administration sets the number of data connections allowed per terminal. There are two types of DCCBs, as follows: • $CDCNET_COMMAND, for interactive connections • $INPUT/$OUTPUT, for batch and interactive connections For each terminal device, the LCM constructs the initial DCCB for the $CDCNET_COMMAND connection. $INPUT/$OUTPUT DCCBs are constructed for terminal devices when connections are created by the terminal user. For batch devices, only one $INPUT/$OUTPUT DCCB is maintained per operational batch device.

Figure 9-3 shows the relationships among the terminal control blocks. There is always one ALCB. Various configurations of the other control blocks may exist within DI memory. The PST type is described earlier in this chapter, under Analyzing DI Hardware Configurations.

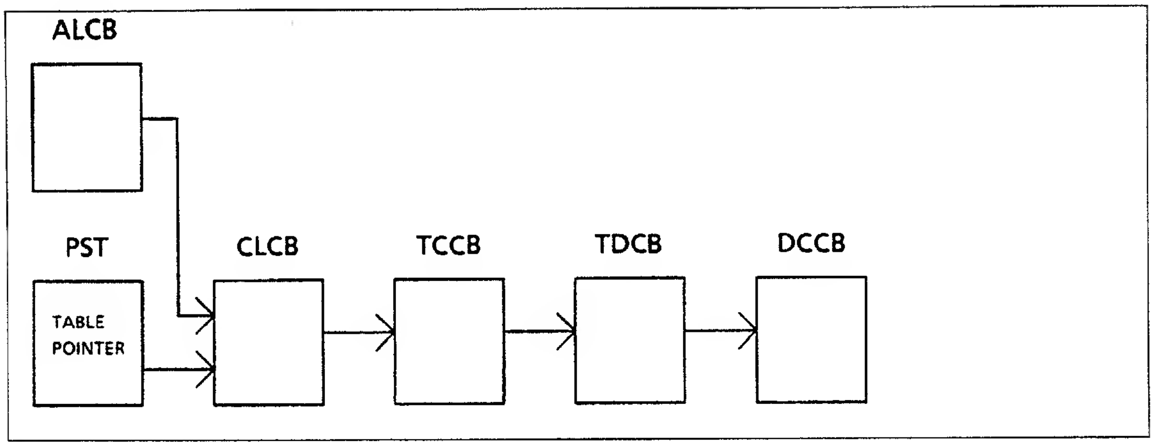


Figure 9-3. Terminal Control Block Relationships

Each of these control blocks has address fields that link it to its owning control block, its owned control blocks, and to the next control block of its type in the chain. The locations of these fields are consistent throughout the five control block types.

Figure 9-4 shows the locations of these fields. The exceptions are for the ALCB, which has neither an owning control block nor any peers, and the DCCB, which owns no control blocks. Bytes 6 through 9 of the DCCB structure point to an output buffer queue.

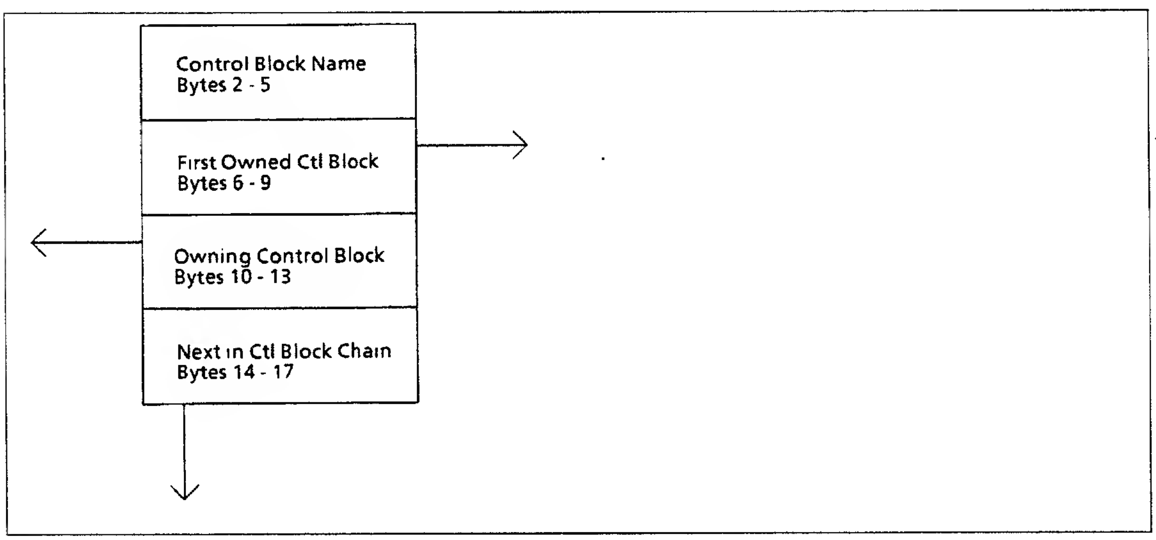


Figure 9-4. Control Block Pointers

The consistent construction of the various control block types results in distinct advantages when you are using the Dump Analyzer. You can use the Dump Analyzer's DISPLAY_LINKED_LIST subcommand to display a chain of one type of control block, or a succession of the terminal control block types.

The complete structure of each line and terminal control block type is described in appendix F.

Analyzing Connections Using NETOU

A NETOU command called DISPLAY_LINE_STATUS (DISLS) causes the DI to display information from its line and terminal control blocks. DISLS must be enclosed in a SENC command, as in the following example:

```
SENC 'DISLS' TDI_A1
```

Output from this command is formatted as follows:

```
FROM TDI_A1                      33010

Line Status
line name                        line       line  tip    line  physical
                                 state      type  name   speed device name
LINE00                           active     ded   async  19200 $LIM0_PORT0
LINE01                           active     ded   async  19200 $LIM0_PORT1
LINE02                           active     ded   async   9600 $LIM0_PORT2
LINE03                           active     ded.  async   9600 $LIM0_PORT3

  .


LINE70                           active     ded   async   9600 $LIM7_PORT0
LINE71                           active     ded   async   9600 $LIM7_PORT1
LINE72                           active     ded   async   9600 $LIM7_PORT2
LINE73                           active     ded   async   9600 $LIM7_PORT3
```

The preceding display is an example of the summary display option, which is the default for this command. DISLS has two other display options: EXPANDED and DETAIL.

The EXPANDED option displays the connected device name(s) for each line. It is selected as in the following example:

```
SENC 'DISLS DO=E' TDI_A1
```

Output from this command is formatted as follows:

```
FROM TDI_A1                      33010

Line Status

LINE00                           tip name:  bsc3270
  device name: REAL_CLUSTER_CONTROLLER          address: 0 /0  state. not ready

LINE01                           tip name:  bsc3270
  device name  $CONSOLE_3000A1_010000           address. 0 /0  state  active

LINE02                           tip name:  bsc3270

LINE03                           tip name:  bsc3270
        .              .
                       .


LINE70                           tip name:  async
  device name: $CONSOLE_3000A1_700000           address: 0 /0  state: active

LINE71                           tip name.  async
  device name: $CONSOLE_3000A1_710000           address. 0 /0  state: active

LINE72                           tip name.  async
  device name  $CONSOLE_3000A1_720000           address: 0 /0  state: active

LINE73                           tip name:  async
  device name. $CONSOLE_3000A1_730000           address: 0 /0  state: active
```

The DETAIL option provides information about each of the connections. It also indicates with a pointer (>) which connection is currently active. The DETAIL option is selected as follows:

```
SENC 'DISLS DO=D' TDI_A1
```

Output from this command is formatted as follows:

```
FROM TDI_A1                     33010

Line Status

REAL_CLUSTER_CONTROLLER            line name. LINE00
  > service name. $CDCNET_COMMAND                     INTERACTIVE
    input state. active     output state: send       output queued   0 /0

$CONSOLE_3000A1_010000             line name: LINE01
  > service name  $CDCNET_COMMAND                     INTERACTIVE
    input state  active     output state: send       output queued.  0 /0
        .            .                                 .
        .                       .                              .
                                                               .

$CONSOLE_3000A1_710000             line name: LINE71
  > service name  $CDCNET_COMMAND                     INTERACTIVE
  input state. active     output state: send       output queued.  0 /0

$CONSOLE_3000A1_720000             line name: LINE72
  > service name  $CDCNET_COMMAND                     INTERACTIVE
    input state  active     output state  send       output queued   0 /0

$CONSOLE_3000A1_7300000000         line name: LINE73
    service name  $CDCNET_COMMAND                     INTERACTIVE
    input state: off        output state: send       output queued.  0 /0
    service name  NOS875907                           INTERACTIVE
    input state  off        output state· hold       output queued   0 /0
    service name· NOS990102                           INTERACTIVE
    input state: off        output state: hold       output queued:  0 /0
    service name: NOS830605                           INTERACTIVE
    input state  off        output state. hold       output queued   0 /0
  > service name: NOS174817                           INTERACTIVE
    input state. active     output state: send       output queued   0 /0
```

Analyzing Connections Using the Dump Analyzer

The Dump Analyzer can be used to locate and display the actual terminal connection control blocks from a DI dump file. It is also equipped with a subcommand, called DISPLAY_LINE_CONTROL_BLOCK (DISLCB), that displays fields from the CLCB for one or for all lines.

DISPLAY_LINE_CONTROL_BLOCKS Subcommand

The following subcommand displays CLCB information from the dump file for LINE00.

```
DISLCB LN=LINE00
```

Output from this subcommand is formatted as follows:

```
CONFIGURED LINE CONTROL BLOCK      AT ADDRESS      1B9712(16)

   POINTER TO FIRST TCCB                           1850B2(16)
   POINTER TO ACTIVE LCB                           1A1470(16)
   POINTER TO NEXT CLCB                            1B33C4(16)
   OPTIONAL TIP EXTENSION POINTER                  29C634(16)
   LINE NAME                                       LINE00
   LINE INTERFACE MODULE                                0
   PORT NUMBER                                          0
   TIP TYPE                                     ASYNC TIP
   LINE TYPE                                     SWITCHED
   FRAMING TYPE                                     ASYNC
   CARRIER TYPE                                  CONSTANT
   LINE SPEED                                       19200
   ASYNC AUTOREC TYPE                                NONE
   CONNECT TIME TIMEOUT                                30
   DISCONNECT TIMEOUT                                  30
   USER CONNECTION LIMIT                                4
   EIA FLOW CONTROL                                 FALSE
   EFC CLOCKING                                         0
   LCSM TASK ID                                     0(16)
   TIP TASK ID                                 1CA9FA(16)
   CONFIGURATION CMD QUEUE                          0(16)
   ADD CB COUNT                                         0
   LCM STATE                     VALUE =            8(16)
   LINE DOWN REASON              VALUE =           FF(16)
   AUTOREC TIP TYPE                             ASYNC TIP
   AUTOREC LINE SPEED                                  26
   AUTOREC CODE SET                                  AUTO
   AUTOREC PARITY                                    MARK
   CONNECT TIMER                                        0
   LCSM LINE ENABLE STATUS                           TRUE
   LCSM STATE                    VALUE =            4(16)
   LCSM AUTOREC STATE            VALUE =            0(16)
```

The following subsections describe how to locate and display the actual line control blocks from a DI dump file. The memory displays can be interpreted using the tables provided in appendix F.

Locating the ALCB

A 10-byte-long ALCB is created in TDI or MTI memory when a line is first configured for the DI. There is an entry point in DI memory, called LCMX, that corresponds with the location of the ALCB.

The ALCB can be displayed using the following Dump Analyzer subcommand:

```
DISM A=LCMX RC=0A
```

The ALCB can also be located by taking the following steps (this is a longer, but perhaps more informative, means of locating the ALCB):

1. Display the first four bytes of the software status service access point (SAP) table using the following Dump Analyzer subcommand:

   ```
   DISM A=SOFTWARE_STATUS_SAP_TABLE RC=4
   ```

   An example of the output from this subcommand follows. These four bytes are a pointer to the first status SAP.

   ```
   STARTING ADDRESS     11F122

   HEX ADDR              HEXADECIMAL DATA                ASCII DATA

     11F122  0010 690E                                    i
   ```

2. Display a linked list of the status SAP entries using the following Dump Analyzer subcommand:

   ```
   DISLL A=first_status_sap_entry LO=0A BC=20 RC=8
   ```

first_status_sap_entry is equal to the address found in the software status SAP table.

Following is an example of the display from this subcommand. Bytes 17 through 31 of each entry is the SAP entry name. Its ASCII representation appears on the right of the display, under the heading ASCII DATA.

```
DISPLAY_LINKED_LIST

START ADDRESS.   10690E

HEX ADDR               HEXADECIMAL DATA                  ASCII DATA

  10690E  0001 0013 7162 0010 7B2A 0010 6C06 000D        qb  {*  l
  10691E  584E 535F 5452 414E 5350 4F52 5443 4B20      OSI_TRANSPORTCK

ADDRESS OF NEXT ELEMENT:    106C06

HEX ADDR               HEXADECIMAL DATA                  ASCII DATA

  106C06  0002 0013 ADF2 0010 7C0E 0010 6F24 000A              _   o$
  106C16  434F 4D4D 414E 445F 4D45 5354 4143 4B20      COMMAND_MESTACK

ADDRESS OF NEXT ELEMENT:    106F24

HEX ADDR               HEXADECIMAL DATA                  ASCII DATA

  106F24  0003 0014 1962 0010 7CF2 0019 2740 0017         b  _   '@
  106F34  4C4F 475F 5355 5050 4F52 545F 4150 504C      LOG_SUPPORT_APPL
```

```
ADDRESS OF NEXT ELEMENT:    192740

HEX ADDR              HEXADECIMAL DATA                        ASCII DATA

  192740  0004 0012 5F98 0019 3B24 001A E632 0007          _  ;$   2
  192750  524F 5554 494E 4700 0000 0007 0000 0015       ROUTING

ADDRESS OF NEXT ELEMENT.    1AE632

HEX ADDR              HEXADECIMAL DATA                        ASCII DATA

  1AE632  0005 001A AC7E 0000 0000 001E BD98 0003            ~
  1AE642  4F53 4100 022C 322C FFDC 4A01 6632 3D7C       OSA  ,2,  J f2=_

ADDRESS OF NEXT ELEMENT     1EBD98

HEX ADDR              HEXADECIMAL DATA                        ASCII DATA

  1EBD98  0006 001E B18E 001E A812 0000 0000 000F
  1EBDA8  4C43 4D5F 4C49 4E45 5F53 5441 5455 5332       LCM_LINE_STATUS2
```

The SAP entry named LCM_LINE_STATUS leads you to the ALCB. Bytes 2 through 5 of this entry record the address of a software status table. The address is 1EB18E(16) in the example.

If there is no LCM_LINE_STATUS entry, the terminal support software has not been loaded into this DI.

Display the first four bytes of the software status table for the LCM_LINE_STATUS SAP entry with the following Dump Analyzer subcommand:

```
DISM A=lcm_software_status_table RC=4
```

lcm_software_status_table is equal to the address found in the LCM_LINE_STATUS SAP entry.

Following is an example of the output from this subcommand. These four bytes indicate the starting address of the ALCB.

```
STARTING ADDRESS    1EB18E

HEX ADDR              HEXADECIMAL DATA                        ASCII DATA

  1EB18E  001D C2EE
```

Once its address is determined, the 10-byte-long ALCB can be displayed using the following Dump Analyzer subcommand:

```
DISM A=alcb_address RC=0A
```

alcb_address is the address found in the software status table for LCM_LINE_STATUS.

Following is an example of the display from this subcommand.

```
START ADDRESS:    1DC2EE

HEX ADDR              HEXADECIMAL DATA                        ASCII DATA

  1DC2EE  0100 414C 4342 0022 43E6                      ALCB
```

The abbreviated name of the control block is recorded in bytes 2 through 5. Bytes 6 through 9 point to the first entry in the CLCB chain. All of the fields in the ALCB record are described in appendix F of this manual.

Locating the CLCB Chain

There is a pointer to the first CLCB in bytes 6 through 9 of the ALCB.

There is also a pointer to the first CLCB for an LCM-controlled port in bytes 20 through 23 of the associated PST entry. To verify that a port is owned by LCM, look for the value 0003 in bytes 18 and 19 of the PST entry in question. For a complete description of the PST and its entries, see the section of this chapter titled Analyzing DI Hardware Configurations Using the Dump Analyzer.

To display the first CLCB associated with the ALCB of the previous example, use the following subcommand:

```
DISM A=2243E6 RC=9A
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS     2243E6

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  2243E6  0202 434C 4342 0028 008A 001F BD2E 0022     CLCB (      =. "
  2243F6  4ABA 0002 42C6 001F 1462 01F2 0403 01C2   J   BF    b r    B
  224406  0000 0201 0000 2580 0202 0101 0100 0004         %
  224416  0002 0000 0000 0000 0028 133E 0000 0000              ( >
  224426  001F BED0 0000 0000 2580 08FF 0200 0200     >P     %
  224436  00C0 0400 0000 0000 0000 0000 0000 0000    @
  224446  0000 0022 43E6 0000 0000 0000 00A6 0093       "Cf          &
  224456  001F 7860 0000 0000 0001 17F4 0000 0000     x         t
  224466  0000 0000 0100 0000 1D7C 0000 0398 0000                |
  224476  2DEC 0000 01F9 0000 0000                    -l    y
```

To display a linked list of four CLCBs, use the following subcommand:

```
DISLL A=first_clcb LO=0F BC=9A RC=4
```

first_clcb is the address of the first CLCB in the linked list.

The abbreviated name of the control block is recorded in bytes 2 through 5. Bytes 6 through 9 point to the first TCCB owned by this CLCB. Bytes 10 through 13 point to the system's ALCB. Bytes 14 through 17 point to the next CLCB in the CLCB chain.

Locating the TCCB Chain

The address of the first in a chain of TCCBs is in bytes 6 through 9 of the CLCB that owns the chain.

To display the first TCCB associated with the CLCB of the previous example, use the following subcommand:

```
DISM A=28008A RC=30
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:   28008A

HEX ADDR            HEXADECIMAL DATA                    ASCII DATA

  28008A   0203 5443 4342 0028 0738 0022 43E6 0000     TCCB ( 8 "Cf
  28009A   0000 0000 0000 0000 0000 0000 0268 0000            h
  2800AA   0000 0000 0000 0000 0000 0000 0000 0000
```

To display a linked list of four TCCBs, use the following subcommand:

```
                                                                 0
DISLL A=first_tccb LO=0F BC=30 RC=4
```

first_tccb is the address of the first TCCB in the linked list.

The abbreviated name of the control block is recorded in bytes 2 through 5. Bytes 6 through 9 point to the first TDCB owned by this TCCB. Bytes 10 through 13 point to the owning CLCB. Bytes 14 through 17 point to the next TCCB in the TCCB chain.

Locating a TDCB Chain

The address of the first TDCB in a chain of TDCBs owned by a TCCB is in bytes 6 through 9 of the owning TCCB.

To display the first TDCB associated with the TCCB of the previous example, use the following subcommand:

```
DISM A=280738 RC=7C
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS     280738

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

  280738  0204 5444 4342 0028 067A 0028 008A 0000     TDCB ( z (
  280748  0000 0000 0000 0000 0000 0000 0000 01E6                  f
  280758  0312 0201 01C2 0000 0000 0028 067A 0028        B      ( z (
  280768  067A 0000 0000 8000 0000 0268 0000 0537    z          h   7
  280778  0000 0036 0000 0451 0000 006F 0000 0001      6   Q    o
  280788  0000 0020 2020 2001 0D20 010A 2001 0C20
  280798  2020 2020 2000 0000 0000 0000 1850 0025                 P %
  2807A8  000A 0D08 1800 0201 0002 0000
```

To display a linked list of four TDCBs, use the following subcommand:

```
DISLL A=first_tdcb LO=0F BC=7C RC=4
```

first_tdcb is the address of the first TDCB in the linked list.

The abbreviated name of the control block is recorded in bytes 2 through 5. Bytes 6 through 9 point to the first DCCB owned by this TDCB. Bytes 10 through 13 point to the owning TCCB. Bytes 14 through 17 point to the next TDCB in the TDCB chain.

Locating a DCCB Chain

The address of the first DCCB in a chain of DCCBs owned by a TDCB is in bytes 6 through 9 of the owning TDCB.

To display the first DCCB associated with the TDCB of the previous example, use the following subcommand:

```
DISM A=28067A RC=56
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS     28067A

HEX ADDR             HEXADECIMAL DATA                   ASCII DATA

  28067A  0205 4443 4342 0026 D94C 0028 0738 0000      DCCB &YL  ( 8
  28068A  0000 0000 0000 0301 0000 0101 0000 0000
  28069A  0000 0000 0000 0100 0000 0000 0000 0000
  2806AA  0000 0000 0000 0000 0028 133E 0205 0000            ( >
  2806BA  4800 8000 00A0 00FF 020D 8D20 2002 0D8D   H
  2806CA  2020 0100 2040                                  @
```

To display a linked list of two DCCBs, use the following subcommand:

```
DISLL A=first_dccb LO=0F BC=56 RC=2
```

first_dccb is the address of the first DCCB in the linked list.

The abbreviated name of the control block is recorded in bytes 2 through 5. Bytes 6 through 9 point to an output buffer queue for this connection. Bytes 10 through 13 point to the owning TDCB. Bytes 14 through 17 point to the next DCCB in the DCCB chain.

Displaying the Output Queue

The address of the data connection output queue is in bytes 6 through 9 of the owning DCCB. Use the Dump Analyzer's DISPLAY_BUFFER_CHAIN subcommand to display the associated data.

To display the output queue associated with the DCCB of the previous example, use the following subcommand:

```
DISBC 26D94C
```

Output from this subcommand is formatted as follows:

```
BUFFER CHAIN DISPLAY

DATA DESCRIPTOR      26D94C(16)     NEXT_DESCRIPTOR      24BF36(16)
NEXT_MESSAGE         2605C6(16)     THE_DATA             26D8B4(16)
OFFSET                   8B(16)     COUNT_BUFFER              5(16)
COUNT_MESSAGE            2A(16)     USAGE_DESCRIPTOR          0(16)

USAGE COUNT               0(16)
DATA AT THE_DATA

  26D8B4  0000  0121  0404  00FF  FF00  2A42  0500  00A2      !      *B
  26D8C4  0108  0025  1000  8303  FF00  00A0  0108  0025      %              %
  26D8D4  3000  C04A  0B80  0049  EA4A  2500  0102  EA02   0  J    I  J%
  26D8E4  ED87  1221  2313  2719  9000  000F  020D  C000      !#  '
  26D8F4  0108  0025  3001  0BE8  212D  4803  0140  0024      %0    !-H   @ $
  26D904  5553  4552  5F50  524F  4301  CC04  0400  FFFF   USER_PROC
  26D914  0033  C012  0835  0611  0900  25FF  FFFF  0014    3   5      %
  26D924  0000  A201  0800  2510  0083  0014  0101  0F00         %
  26D934  0000  2455  5345  525F  5052  4F43  [0001  0000    $USER_PROC
  26D944  2E]                                              .


DATA DESCRIPTOR      24BF36(16)     NEXT_DESCRIPTOR           0(16)
NEXT_MESSAGE              0(16)     THE_DATA             24BE9E(16)
OFFSET                    1(16)     COUNT_BUFFER             25(16)
COUNT_MESSAGE             0(16)     USAGE_DESCRIPTOR          0(16)

USAGE COUNT               0(16)
DATA AT THE_DATA:

  24BE9E  0000  [0A0D  4578  7065  6374  696E  6720  636F       Expecting co
  24BEAE  6D6D  616E  642C  2066  6F75  6E64  2048  495F   mmand, found HI_
  24BEBE  4A4F  4E41  533A  2E]                            JONAS:.


DATA DESCRIPTOR      2605C6(16)     NEXT_DESCRIPTOR           0(16)
NEXT_MESSAGE         2714F8(16)     THE_DATA             22D370(16)
OFFSET                   7B(16)     COUNT_BUFFER             15(16)
COUNT_MESSAGE            15(16)     USAGE_DESCRIPTOR          0(16)

USAGE COUNT               D(16)
DATA AT THE_DATA:

  22D370  0000  03C6  0404  00FF  FF00  2AC4  0500  00A2              *
  22D380  0108  0025  1000  8303  FD00  0000  6308  0025      %         c  %
  22D390  3001  6161  9980  00FE  D34D  4F00  0100  0200   0 aa      MO
  22D3A0  0925  FFFF  FF00  0000  0000  0000  0000  0000    %
  22D3B0  0000  0000  0000  0000  0000  0000  0000  0000
  22D3C0  0000  0000  0000  0000  0000  0000  0000  0000
  22D3D0  0001  E201  E404  0400  FFFF  002A  5010  4C01               *P L
  22D3E0  A004  0400  FFFF  0043  C005  0DC0  [0001  0000            C
  22D3F0  2E49  6E70  7574  2064  6973  6361  7264  6564   .Input discarded
  22D400  2E]                                              .
     .      .      .            .      .      .      .      .
     .      .      .      .     .      .      .      .      .
     .      .      .      .     .      .      .      .
```

Analyzing Connections Using NPA

NPA generates statistical reports about connections and terminals. These report types are summarized in table 9-6. Examples of all of the following reports can be found in chapter 6, NPA Reports and Report Formats.

Table 9-6. NPA Reports

NPA Report	Description
CONNRP1	Hourly connection report on the number of connections initiated and terminated, by service.
CONNRP2	Daily connection report on the number of connections initiated and terminated, by service.
TELNRP1	Hourly TELNET connection report.
TELNRP2	Daily TELNET connection report.
TERMRP1	Hourly terminal report on block input and output.
TERMRP2	Hourly terminal statistics report on characters input and output.
X25CRP1	X.25 connection statistics report sorted by DI and time.
X25CRP2	X.25 connection statistics report sorted by service name and DI on a daily basis.

Remote Line Monitor Utility (RLM) 10

Remote Line Monitor Utility (RLM) 10

The NOS/VE Remote Line Monitor records and/or displays all received and transmitted characters on a LIM and port that are supported by the standard CDCNET CIM Firmware and that use protocols defined for Remote Line Monitor (only ASYNC and HASP protocols have been defined).

Remote Line Monitor has two parts, the Remote Line Monitor utility, a NOS/VE application, and the CDCNET Remote Line Monitor TIP. Only the NOS/VE application is discussed in this chapter.

Commands to define and delete the NOS/VE application are included in MANAGE_APPLICATION_DEFINITIONS and requires NETWORK_APPLICATION_MANAGEMENT capability.

You must be validated to use the NETWORK_OPERATOR_UTILITY to use the Remote Line Monitor. To use the Remote Line Monitor, enter the following commands:

```
CREATE_COMMAND_LIST_ENTRY ..
  ENTRY=:$SYSTEM.$SYSTEM.REMOTE_LINE_MONITOR.LMF$COMMAND_LIBRARY
REMOTE_LINE_MONITOR
```

Only one communication line can be monitored at a time in a Remote Line Monitor session. Only one line at a time (a CDCNET Remote Line Monitor TIP restriction) may be monitored in any given DI.

Starting a Remote Line Monitor Session

The following command starts a Remote Line Monitor session.

REMOTE_LINE_MONITOR, REMLM, RLM
STATUS = status variable

STATUS

The normal NOS/VE status parameter.

Main Menu Screen

When the REMOTE_LINE_MONITOR starts, it displays the Main Menu screen which shows the various functions available for a Remote Line Monitor session. See figure 10-1.

```
                              REMOTE LINE MONITOR            Lines 1 to 19 of 19

        FUNCTION KEYS                              CURRENT SETTINGS
     ------------------------------             ------------------------------

     MAINTENANCE                                Monitored System = AHR_TDI_300784
                                                Monitored Lim    = 3
      SF1 to - Setup                            Monitored Port   = 1
       F1 to - Manage files
                                                Forward Timer  = 2000
     MONITORING
                                                Line Protocol  = ASYNC
       F2 to - Record
       F3 to - Display                          Display Format = ASCII
       F4 to - Display and Record
                                                Display Width  = 80
     ANALYSIS

       F7 to - Format and Edit
       F8 to - Edit                             { Use Setup to change settings. }
        Setup
 f1  FilMgt f2 Record f3 Dis  f4 DisRec f5 HELP  f6 Quit  f7 FormEd f8 Edit
```

Figure 10-1. Main Menu Screen

The following table describes the function keys available on the Main Menu screen.

Key	Description
Shift F1	Displays the Setup screen where you set values to indicate the line to monitor, the forwarding timeout value for the Remote Line Monitor TIP, the protocol in use on the monitored line, the display format used for formatting line data, and the page width for formatted data. See Setup Screen section later in this chapter.
F1	Starts the NOS/VE File Manager. See File Management section later in this chapter.
F2	Starts monitoring a remote line and writes the unformatted data to a file. See Record section later in this chapter.
F3	Starts monitoring a line and displays the formatted data to the terminal. See Display section later in this chapter.
F4	Starts monitoring a line and displays the formatted data to the terminal and writes the unformated data to a file. See Record and Format and Edit sections later in this chapter for file naming conventions.
F5	Displays a help screen with a row of function keys. Press the corresponding keyboard function key for the subject about which you want help; this displays a help screen for that function. Some of the function keys have both brief and full help available.
F6	Ends the Remote Line Monitor session. Terminates the utility.
F7	Formats a file of recorded line data and starts a NOS/VE File Editor session with the formatted result file. See Format and Edit section later in this chapter.
F8	Starts a NOS/VE File Editor session with a formatted result file. See Edit section later in this chapter.

Screens

The following subsections describe the screens that you can go to by using the function keys on the Main Menu screen.

Keys not under the direct control of the REMOTE_LINE_MONITOR are defined by various screen interfaces that the Remote Line Monitor uses and are not discussed in this chapter. If you need help with these keys, refer to Creating Interactive Procedures and Utilities section of the NOS/VE System Usage manual, or use the HELP key while in these screen interfaces to obtain help.

Setup Screen (Setup)

The Setup screen (figure 10-2) is where you set the values for controlling the remote line monitoring and evaluating the data. This screen is the first screen displayed when you start a Remote Line Monitor session if you have not previously used Remote Line Monitor and/or if Monitored System=UNKNOWN. You can reach this screen by pressing the Setup function key (Shift F1) on the Main Menu screen.

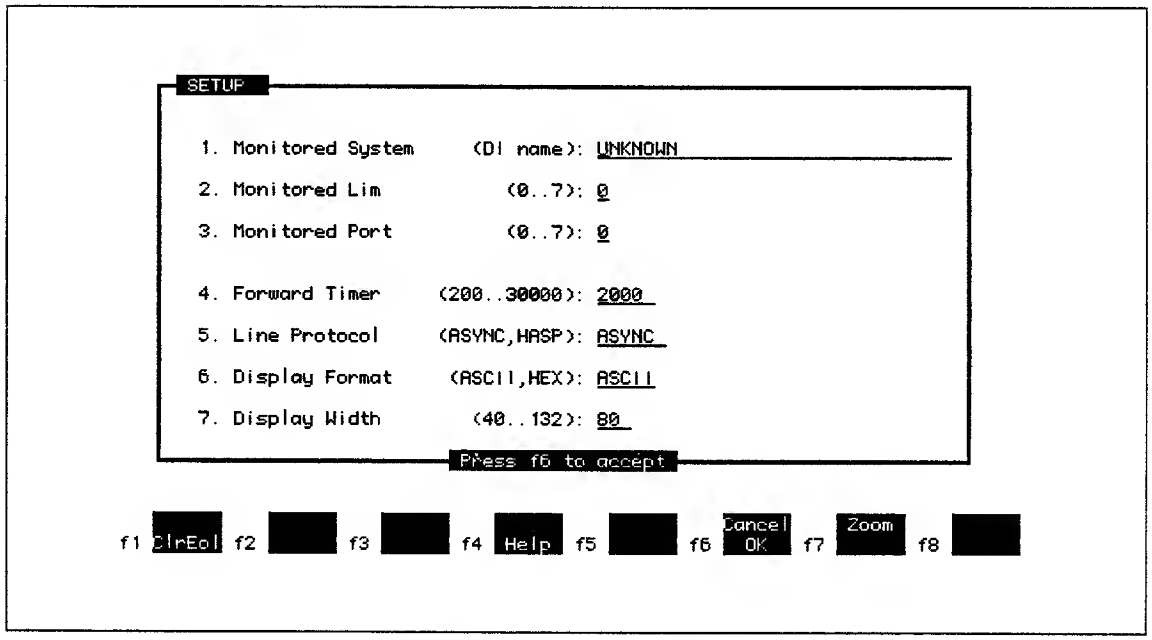


Figure 10-2. Setup Screen

The following table describes the values of the current settings on the Setup screen:

Value	Description
Monitored System	The name of the monitored DI.
Monitored Lim	The LIM number on the DI with the monitored port.
Monitored Port	The port number on the LIM.
Forward Timer	The time period, specified in milliseconds, that is to be used by the Remote Line Monitor TIP as the criterion for deciding when to forward buffered data. The Remote Line Monitor TIP forwards monitored data to the Remote Line Monitor NOS/VE application whenever either the timer specified by this parameter expires or when the TIP's internal buffer, approximately 1400 bytes, is full. Decreasing the Forward Timer value increases the frequency at which data packets are forwarded. You might want to do this to increase the apparent real-time updating of your display or to obtain more frequent millisecond clock timings in the recorded data. However, this can cause congestion on the network by increasing the number of network packets on the network. Increasing the Forward Timer value decreases the frequency at which data packets are forwarded and increases the amount of data in the packets (until the buffer limit is reached). You might want to do this when you don't require precise timing information and when you want to prevent congestion on the network and minimize the loss of data due to flow control. If most data packets are forwarded because the TIP buffer is full, then changing the Forward Timer value has little or no effect unless decreased below the time it takes to fill the TIP buffer.
Line Protocol	The protocol in use on the monitored line. This corresponds to the line TIP. This is not verified by the utility and affects only the formatting of data.
Display Format	The format in which monitored data is displayed to the terminal or formatted to a result file.
Display Width	The maximum number of columns of formatted data displayed to the terminal or formatted to a result file. The number of characters displayed may not be as wide as specified since symbolic characters and hex characters are not broken across lines.

Values set on this screen are saved until changed, including between Remote Line Monitor sessions. These values and others maintained internally by the utility are saved in file $USER.$REMOTE_LINE_MONITOR.CONFIGURATION.

File Management (FilMgt)

The File Management function key enters the NOS/VE File Manager so you can maintain your Remote Line Monitor files. See figure 10-3.

When NOS/VE File Manager starts, it is positioned in the catalog hierarchy as close to the current files being used as the Remote Line Monitor can deduce from the environment. If a system is known, then the catalog containing files for that system is shown; otherwise, the $USER.$REMOTE_LINE_MONITOR catalog is shown. Function key F6 (Quit) ends the File Manager session and returns you to the Remote Line Monitor Main Menu.

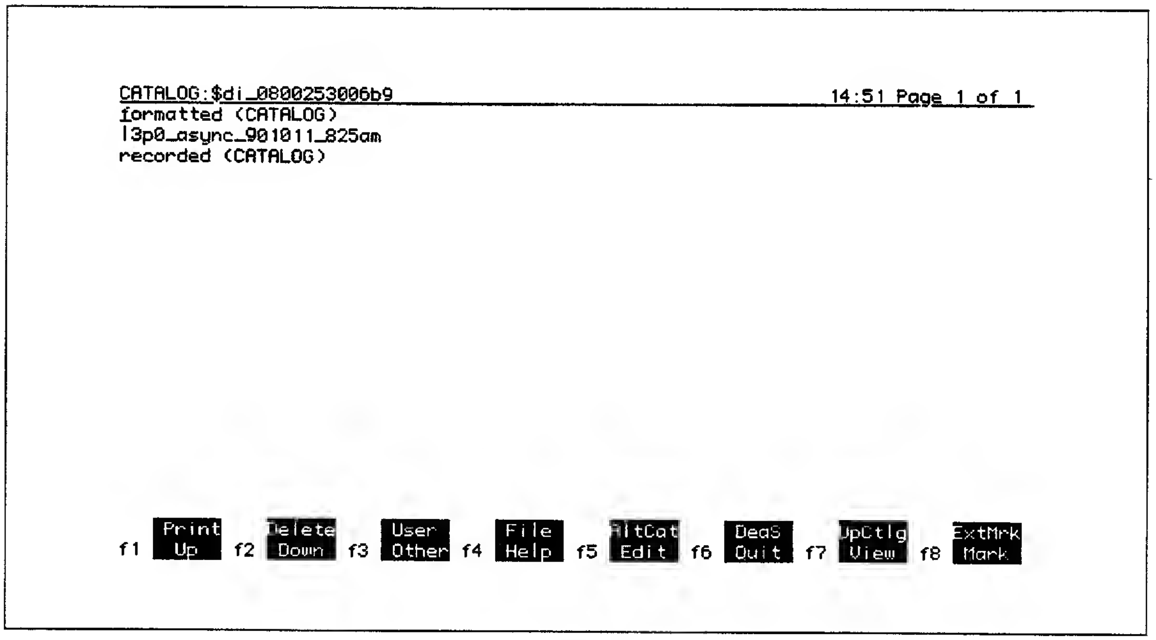


Figure 10-3. File Management Screen

The following shows an example of the Remote Line Monitor catalog structure:

```
$bcm {user}
  $remote_line_monitor
    ahl_tdi_3006b9 { system name }
      formatted
        12p3_hasp_900222_1005am_hex
        12p3_hasp_900222_1009am_ascii
        12p3_hasp_900224_1021am_hex
        12p3_hasp_900224_1024am_hex
        13p0_async_900108_1005am_ascii
        13p0_async_900108_1005am_hex
      recorded
        12p3_hasp_900222_1009am
        12p3_hasp_900224_1021am
        12p3_hasp_900224_1024am
        12p3_hasp_900224_1032am
        13p0_async_900108_1005am
    configuration { settings saved between remote line monitor sessions }
    rlm_tdi { system name }
      formatted
        12p3_hasp_900222_1005am_hex
        12p3_hasp_900222_1009am_ascii
        13p0_async_900108_1005am_ascii
        13p0_async_900108_1005am_hex
      recorded
        12p3_hasp_900222_1009am
        13p0_async_900108_1005am
```

Record (Record)

The Record function key starts monitoring a remote line as specified by the system, LIM, and port values from the Setup screen and writes the unformatted data to a file.

The recorded data file is located in catalog:

$USER.$REMOTE_LINE_MONITOR.*system*.RECORDED

where *system* is the value of system from the Setup screen.

The file is named L*lim*P*port_protocol_date_time*.

where *lim*, *port*, and *protocol* are values from the Setup screen. Date and time are determined at the time the Record function starts. File naming is done automatically by the Remote Line Monitor; you cannot change or alter it. However, you can change the file names later using File Management.

While data is recorded, the Remote Line Monitor displays the size of the file receiving the data. Function key F6 (Stop) terminates recording and returns you to the Remote Line Monitor Main Menu.

Each time the file size is updated on the display, the time (displayed on the title line) is also updated. The refresh rate is approximately 10 seconds.

Display (Dis)

The Display function key starts monitoring a remote line and displays formatted data at your terminal. The data format is determined by values set on the Setup screen. Default is ASCII.

As soon as the connection between the Remote Line Monitor on NOS/VE and the Remote Line Monitor on the DI is established, the DI sends an elapsed time of 0 millisecond with no data. Your screen clears and displays <0 milliseconds elapsed>, letting you know that NOS/VE and the DI are communicating. This is important because it lets you know that the connection is made in cases where there is no data on the line being monitored.

To stop the display and return to the Main Menu screen, enter a terminate break.

NOTE

Remote Line Monitor is in screen mode and only an Attention Character with ATTENTION_CHARACTER_ACTION of 2 to 9 or a Break Key with a BREAK_KEY_ACTION of 2 to 9 terminates the display and returns you the Main Menu screen.

In the event that both ATTENTION_CHARACTER_ACTION and BREAK_KEY_ACTION have values outside the range of 2 to 9, the sequence of <break><ctrl-x>%2 performs the same function.

To use the hold page function to control viewing displayed data, you must select the hold page attributes before starting the display function (you can do this from the home line or before entering Remote Line Monitor).

See Data Integrity later in this chapter for a discussion of minimizing potential for lost data when using the display function. See Formatted Data later in this chapter for examples and explanations of formatted data.

Display and Record (DisRec)

The Display and Record function key works the same as the Display function key, except that it records unformatted data to a file and displays formatted data to the terminal.

Format and Edit (FormEd)

The Format and Edit function key formats a file and then starts a NOS/VE File Editor session so you can edit the file. You begin by deciding which recorded file to format from a file selection screen. The file selection screen is positioned at the last recorded data file. If no last recorded data file exists or the Setup screen has been changed, then the file selection screen is positioned at the first file in the catalog containing the recorded data files for the specified system. If the specified system catalog does not exist, then the file selection screen shows the $USER.$REMOTE_LINE_MONITOR catalog. Use the UpCtlg and View functions of the file selection screens to find and select different recorded data files. Position cursor on the desired file and press function key F6 (Accept) to select the file.

After you press function key F6, the utility begins formatting. While the file is formatting, a display continuously shows the current size of the file being formatted. When formatting completes, a NOS/VE File Editor session begins using the formatted file. Enter function key F6 (Stop) to terminate formatting before all data has been processed and to return you to the Main Menu screen.

NOTE

The formatted file is approximately twice the size of the recorded file.

The formatted data file is located in catalog:

$USER.$REMOTE_LINE_MONITOR.*system*.FORMATTED

where *system* is the value of system from the Setup screen.

The file is named .LlimPport_protocol_date_time_*format*.

where *format* is the display format from the Setup screen appended to the recorded file name.

Each time the file size is updated on the display, the time (displayed on the title line) is also updated. The refresh rate is approximately 10 seconds.

The formatted file name is the same as the recorded file name, except that the formatted file is located in the catalog FORMATTED, and a display format extension (ASCII or HASP) has been added to the end of the name. See figure 10-4.

See Data Integrity later in this chapter for a discussion of minimizing lost data when using the display function.

See Formatted Data later in this chapter for examples and explanations of data.

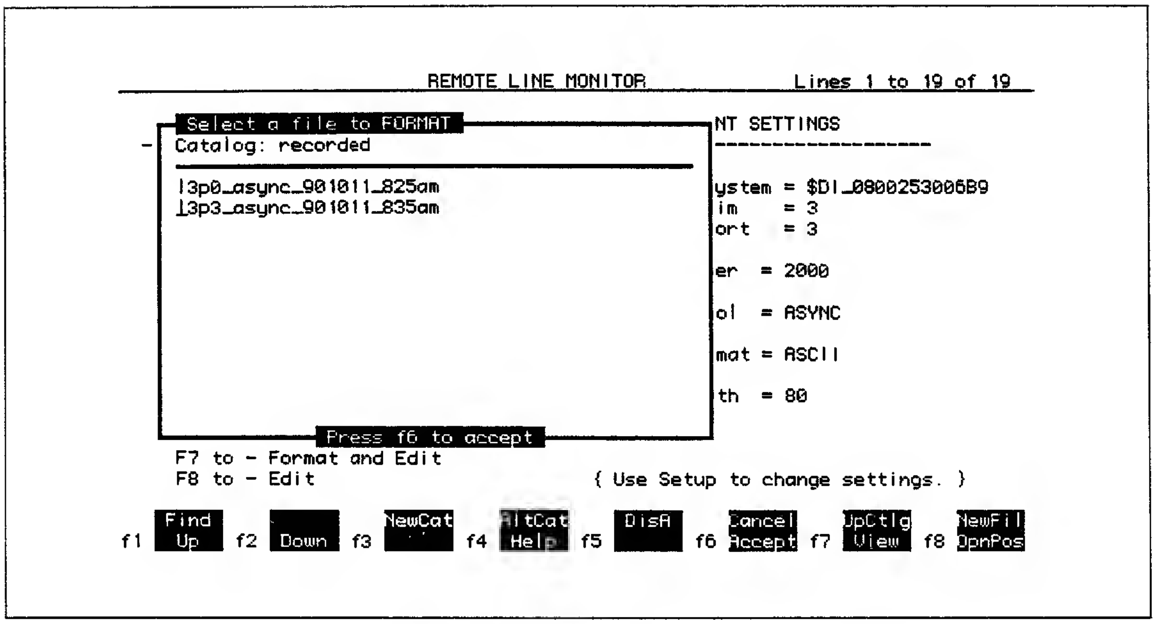


Figure 10-4. Format and Edit Screen

Edit (Edit)

The Edit function key is similar to Format and Edit, except that the file selection screen is initially positioned at the most recently formatted file. If no formatted data file exists or the Setup screen has been changed, then the file selection screen is positioned at the first file on the catalog containing the formatted data files for the specified system. If the specified system catalog does not exist, then the file selection screen shows the $USER.$REMOTE_LINE_MONITOR catalog. Position to the desired file, then enter function key F6 (Accept) to start a NOS/VE File Editor session using the formatted file. Function key F7 (View) is a read-only display and is part of the NOS/VE file Generic Screen interface. See figure 10-5.

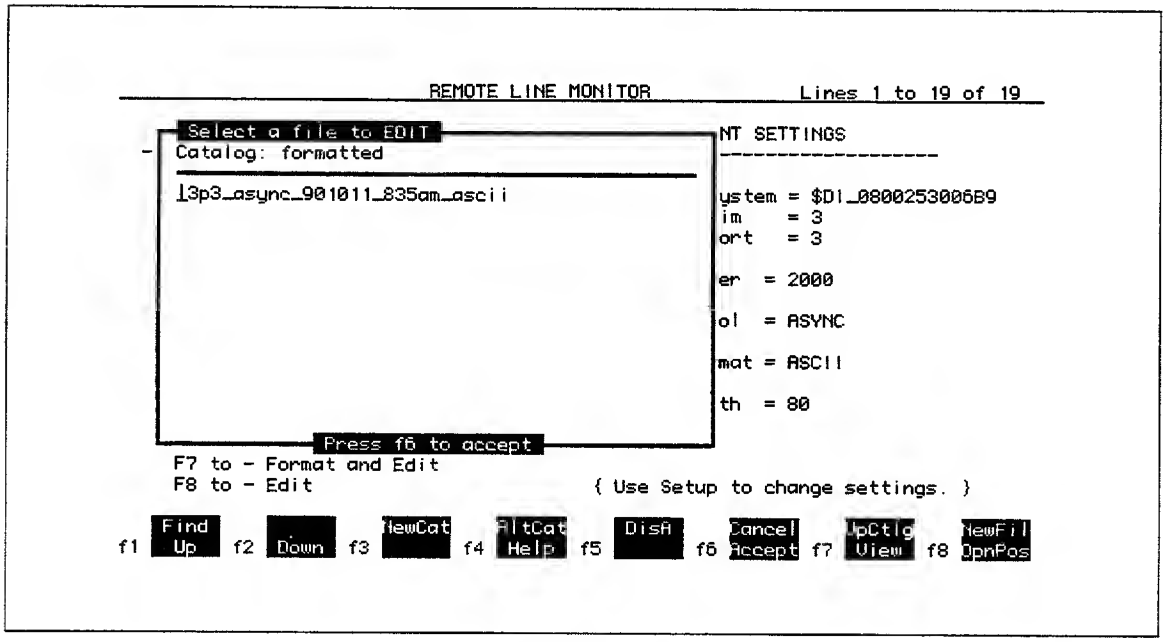


Figure 10-5. Edit Screen

Data Integrity

If losing data would be a problem for you, then you should configure Remote Line Monitor as efficiently as possible to optimize data capture. The following table lists some possible reasons for loss of data and actions you can take to prevent or minimize loss of data.

Reason	Solutions
Use of Hold Page.	Hold Page = OFF.
	Use Record function.
Use of Ctrl-S, Ctrl-Q.	Use Record function.
Monitoring terminal line speed less than monitored line speed.	Use a monitoring terminal speed >= to monitored line speed.
	Use Record function.
Large volume of data traffic being monitored.	Use a monitoring terminal line speed greater than monitored line speed.
	Use Record function.
Remote Line Monitor competing for VE host resources.	Increase application priority using application scheduling (see NOS/VE System Performance and Maintenance manual, volume 1).
	Use Record function.
Busy/congested network.	Monitor at a less busy time.

The first four reasons assume you are displaying the data. The last two reasons apply to all monitoring modes.

The best way to minimize any type of data loss is to record unformatted data to disk.

NOTE

Do not increase the user's job priority; doing so can impact overall system performance. Use application scheduling to ensure the application gets higher priority with a limit on CPU use.

Data Formats

Remote Line Monitor uses unformatted and formatted data. The differences are discussed in the following sections.

Unformatted Data (Recorded)

Data is written (recorded) to a system-specified variable record type file exactly as it is received from the Remote Line Monitor application on the DI. The first line of each recorded data file contains the date and time that the first data was received. Each block of data is written as a variable length record. Each record contains the data header from the DI application followed by up to 1400 bytes of line data.

Formatted Data

Data is formatted as a result of interactive viewing (Dis and DisRec functions) and as a result of formatting (FormEd function) previously recorded data. Data is formatted to the number of columns specified by the display width value on the Setup screen.

Formatting is based on values set using the Setup function. Display Format determines how characters are displayed (ASCII or HEX). Line Protocol determines what character translation is performed when displaying ASCII. Display Width determines the number of columns used for page width for formatted data.

In a formatted result file, the first line of the file is a string identifying the file as FORMATTED REMOTE LINE MONITOR DATA, and the second line is the file name of the recorded data file which was formatted (this line may wrap to the third line if the name is longer than the page width). The third line indicates the time recording was initiated and serves as an approximate reference point for the milliseconds elapsed counters throughout the rest of the formatted data.

In data displayed to the terminal (using Display or Display and Record), the first three lines of information are not displayed.

Display Formats

In the following examples (figures 10-6 through 10-9), the paired lines starting with:

```
--I>
  O>
```

indicate simultaneous input and output. The examples have been formatted using a 60-character page width.

Data is displayed as it is processed by the DI (full duplex must be processed synchronously by the DI one character at a time). Either the input character or the output character is the monitored character (never both). The character on the corresponding input or output representation is a placeholder only, not data.

Each block is preceded with the milliseconds elapsed since the first block was received. Lost data indicates the number of characters lost between the preceding block and the following block. The milliseconds elapsed following lost characters indicates the time the first character of the actual data that follows was recorded.

Certain escape characters embedded in the data by the DI are displayed symbolically inside < > when formatted (for example, <CRC> and <bad CRC>). This allows use of new escape characters for the DI code without breaking this application. When an unknown escape character is found, it is formatted as <?hex_number>.

ASYNC-ASCII Display Format

Figure 10-6 show an example of the ASYNC-ASCII display format. ASYNC protocol is displayed using the ASCII display format.

Periods (.) and colons (:) act as placeholders on the line opposite the processed data. A period indicates a character with the parity bit set to 0. A : (colon) indicates a character with the parity bit set to 1. Blanks indicate that the data on the other line represents one character, starting from the placeholder (for example, when displaying an unprintable character or symbol, only the first character has a placeholder on the opposite line).

On the input and output lines, the values inside < > (angle brackets) are ASCII symbols, or HEX values for nondisplayable characters (greater than 127), or symbolically displayed escape sequences (see examples later in this section).

The following example shows the DISPLAY_CATALOG command followed by a carriage return on the input line and then a line feed on the output line. Note that the parity is for the symbolically displayed character, not the leading < (angle bracket).

```
--I>DISC<CR>.
  O>:.:.:    <LF>
```

If either . (period) or : (colon) is displayed on both lines (a very rare occurrence), then you need to format the data in HEX display format to determine whether the character was input or output.

```
FORMATTED REMOTE LINE MONITOR DATA
FILE :PEWTER.BCM.$REMOTE_LINE_MONITOR.RLM_TDI.RECORDED.L2P2_
ASYNC_900212_250PM
Remote Line Monitoring began February 12, 1990 14:50:14.447


   < 0 milliseconds elapsed>


   < 55470 milliseconds elapsed>

--I>DISV $TIME<CR>:   .    ..::.:.:.   :   .DISV
  O>...........    <LF><CR>14:51:13<CR><LF>/....


   < 59530 milliseconds elapsed>

--I> $DATE<CR>:   .   .::::::...   :   .
  O>.......    <LF><CR>1990-02-12<CR><LF>/


   < 66790 milliseconds elapsed>

--I>DISC
  O>......


   < 71510 milliseconds elapsed>

--I>DO=F<CR>:   .   :....:::::..::..   :   ....::::.......
  O>.....    <LF><CR>$FILE_MANAGER_BF<CR><LF>    NUMBER OF C

--I>:...::.........:...::.:.:...:.......:.:.:..   :   ::...
  O>YCLES:      1, ACCOUNT: NONE, PROJECT: NONE<CR><LF>ASCII

--I>:..:::...   :   ....:::.......:...::........:...::.:.:
  O>_EXAMPLE<CR><LF>    NUMBER OF CYCLES:      1, ACCOUNT: N

--I>.:...:.......:.:.:..   :   :::.::..   :   ....::::.....
  O>ONE, PROJECT: NONE<CR><LF>ASYNCASC<CR><LF>    NUMBER OF
```

Figure 10-6. ASYNC-ASCII Display

ASYNC-HEX Display Format

Figure 10-7 shows an example of the ASYNC-HEX display format. HEX displays are very similar except all characters are displayed in HEX and the only characters displayed inside < > are DI escape codes. No parity is indicated since all 8 bits are displayed in the HEX values. This example uses the same data as the ASYNC-ASCII examples. The ASCII values for characters (both input and output) are displayed on the third line of each input/output set.

```
FORMATTED REMOTE LINE MONITOR DATA
FILE :PEWTER.BCM.$REMOTE_LINE_MONITOR.RLM_TDI.RECORDED.L2P2_
ASYNC_900212_250PM
Remote Line Monitoring began February 12, 1990 14:50:14.447


   < 0 milliseconds elapsed>


   < 55470 milliseconds elapsed>

--I>44495356202454494D450D. . . . . . . . . . . . . . 44495356
  O>. . . . . . . . . . . . . 8A0D3134BAB531BA31B30D8A2F. . . .
    D I S V   $ T I M E      1 4 : 5 1 : 1 3    / D I S V

   < 59530 milliseconds elapsed>

--I>2024444154450D. . . . . . . . . . . . . . . .
  O>. . . . . . . . 8A0D31B9B9B0ADB032AD31320D8A2F
      $ D A T E      1 9 9 0 - 0 2 - 1 2    /

   < 66790 milliseconds elapsed>

--I>444953432020
  O>. . . . . .
    D I S C

   < 71510 milliseconds elapsed>

--I>444F3D460D. . . . . . . . . . . . . . . . . . . . . . . .
  O>. . . . . 8A0DA446494C45DFCDC1CEC1C74552DFC2460D8A202020
    D O = F      $ F I L E _ M A N A G E R _ B F
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>20CED5CDC24552204F462043D9434C45D3BA2020202020312C20C143
      N U M B E R   O F   C Y C L E S :         1 ,   A C
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>434FD5CE54BA20CE4FCE452C20D0524F4A454354BA20CE4FCE450D8A
    C O U N T :   N O N E ,   P R O J E C T :   N O N E
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>C1D3434949DF4558C1CDD04C450D8A20202020CED5CDC24552204F46
    A S C I I _ E X A M P L E           N U M B E R   O F
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>2043D9434C45D3BA2020202020312C20C143434FD5CE54BA20CE4FCE
      C Y C L E S :          1 ,   A C C O U N T :   N O N
```

Figure 10-7. ASYNC-HEX Display

HASP-ASCII Display Format

Figure 10-8 shows an example of the HASP-ASCII display format.

The display is processed identically to ASYNC-ASCII, except that all characters are translated to EBCDIC before they are displayed and all 8 bits are valid so no parity is indicated. No special formatting is done for the HASP protocol.

NOTE

EBCD26 and EBCD29 character sets for HASP are incorrectly translated as EBCDIC when formatted for ASCII display; however, you can see untranslated EBCD26 and EBCD29 values by setting the display format to HEX.

```
FORMATTED REMOTE LINE MONITOR DATA
FILE :PEWTER.BCM.$REMOTE_LINE_MONITOR.RLM_TDI.RECORDED.L2P3_
HASP_900213_355PM
Remote Line Monitoring began February 13, 1990 15:55:40.240


   < 0 milliseconds elapsed>


   < 330 milliseconds elapsed>

--I><SYN><SYN><DLE><ACK0>.    .    .    .    .    .........
  O>.    .    .    .     <SYN><SYN><SYN><DLE><STX>  {ja :PE

--I>.........    .    .    .    .    .    <SYN><SYN><DLE>
  O>WTER.BPF<NUL><NUL><DLE><ETB><CRC><PAD>.    .    .

--I><ACK0>.    .    .    .    .    .    .    .    .    <SYN>
  O>.    <SYN><SYN><SYN><DLE><ACK0><PAD><PAD><PAD>.

--I><SYN><DLE><ACK0>.    .    .    .    .    .    .    .
  O>.    .    .    <SYN><SYN><SYN><DLE><ACK0><PAD><PAD>

--I>.    <SYN><SYN><DLE><ACK0>.    .    .    .    .
  O><PAD>.    .    .    .    <SYN><SYN><SYN><DLE><ACK0>

--I>.    .    .    <SYN><SYN><DLE><ACK0>.    .    .
  O><PAD><PAD><PAD>.    .    .    .    <SYN><SYN><SYN>
```

Figure 10-8. HASP-ASCII Display

(Continued)

(Continued)

```
--I>.     .     .     .     .      <SYN><SYN><DLE><ACK0>.
  O><DLE><ACK0><PAD><PAD><PAD>.    .     .     .     <SYN>

--I>.     .     .     .     .     .     .     <SYN><SYN><DLE>
  O><SYN><SYN><DLE><ACK0><PAD><PAD><PAD>.     .     .

--I><ACK0>.     .     .     .     .     .......    ....    ..
  O>.     <SYN><SYN><SYN><DLE><STX>  {jab<NUL>jab<NUL>ja

--I>.     ...............................................
  O><PAD>display_catalog                               NOS

--I>...............................     ..................
  O>/VE  a9180  SJVL              <NUL>ja4

--I>....................................    ..............
  O>  1990-02-13     15:55:42     PAGE  1<NUL>jaLCATALOG :pe

--I>.........    ....    ..................    ...........
  O>wter.bpf<NUL>jab<NUL>ja    FILE: CFG140<NUL>jaJ   FILE:

--I>..........    ...................    .................
  O> CONF_138<NUL>ja}   FILE: CONF_LM<NUL>jaL    FILE: DISC_

--I>......    ...............    ..................    ...
  O>$USER<NUL>ja    FILE: EL<NUL>ja    FILE: PROLOG<NUL>jaM

--I>....................    ............................
  O>   FILE: RLM_DI_8103<NUL>ja    FILE: RLM_MESSAGE_TEMPLA

--I>....    .............................    .    .
  O>TES<NUL>ja    FILE: SCU_EDITOR_EPILOG<NUL><NUL><DLE>

--I>.     .     .     <SYN><SYN><DLE><ACK0>.     .     .
  O><ETB><CRC><PAD>.     .     .     .     <SYN><SYN><SYN>
```

Figure 10-8. HASP-ASCII Display

HASP-HEX Display Format

Figure 10-9 shows an example of the HASP-HEX display format. The HEX display format leaves all HASP characters untranslated. The display is identical to an ASYNC-HEX display.

```
FORMATTED REMOTE LINE MONITOR DATA
FILE :PEWTER.BCM.$REMOTE_LINE_MONITOR.RLM_TDI.RECORDED.L2P3_
HASP_900213_355PM
Remote Line Monitoring began February 13, 1990 15:55:40.240


   < 0 milliseconds elapsed>


   < 330 milliseconds elapsed>

--I>32321070. . . . . . . . . . . . . . . . . . . . . . . . .
  O>. . . . 32323210028D80C09181CB7AD7C5E6E3C5D94BC2D7C60000
                          { j a   : P E W T E R . B P F
--I>. . .    . 32321070. . . . . . . . . 32321070. . . . . .
  O>1026<CRC>FF. . . . 3232321070FFFFFF. . . . 3232321070FF

--I>. . 32321070. . . . . . . . 32321070. . . . . . . . . 3232
  O>FFFF. . . . 3232321070FFFFFF. . . . 3232321070FFFFFF. .

--I>1070. . . . . . . . 32321070. . . . . . . . . . . . .
  O>. . 3232321070FFFFFF. . . . 32323210028E80C0918182009181
                                                { j a b   j a
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>82009181FF8489A2979381A8608381A38193968740404040404040404040
    b   j a   d i s p l a y _ c a t a l o g
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>40404040404040404040404040404040404040404040404040D5D6E261
                                                     N O S /
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>E5C5404081F9F1F8F04040E2D1E5D340404040404040404040404040
    V E     a 9 1 8 0     S J V L
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>4040009181F440404040404040404040404040404040404040F1F9F9F0
          j a 4                                      1 9 9 0
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>60F0F260F1F34040404040F1F57AF5F57AF4F24040404040D7C1C7C5
    - 0 2 - 1 3           1 5 : 5 5 : 4 2            P A G E
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>40F1009181D3C3C1E3C1D3D6C7407A9785A6A385994B829786009181
      1   j a L C A T A L O G   : p e w t e r . b p f   j a
--I>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
  O>82009181CF404040C6C9D3C57A40C3C6C7F1F4F0009181D1404040C6
    b   j a       F I L E :   C F G 1 4 0   j a J       F
```

Figure 10-9. HASP-HEX Display

Security

All uses of the Remote Line Monitor are recorded to discourage misuse. On NOS/VE, information is written to :$SYSTEM.$SYSTEM.REMOTE_LINE_MONITOR_USAGE. Users should only be allowed to append data to this file. The following is an example of data recorded for one monitoring session:

```
Remote Line Monitoring starting:  02/15/90  14:15:01
    User=BCM
    From family PEWTER
    Monitoring - LIM=2  Port=3  System=RLM_TDI
Remote Line Monitoring ending:    02/15/90 14:15:08
    User=BCM
    From family PEWTER
    Monitoring - LIM=2  Port=3  System=RLM_TDI
    Total data received =  898
```

The Remote Line Monitor TIP also logs similar information in the DI logs.

Cancelling the DI Remote Line Monitor

In the unlikely event that the NOS/VE Remote Line Monitor terminates after it sends a DEFINE_REMOTE_LINE_MONITOR to the DI and before it establishes a connection to the Remote Line Monitor TIP, you must enter NETOU and send the CANCEL_REMOTE_LINE_MONITOR command to the DI. Until you do this, it appears to NOS/VE that there is already a Remote Line Monitor session in progress and NOS/VE informs you that REMOTE_LINE_MONITOR is already defined.

CAUTION

Do not cancel another user's Remote Line Monitor session.

Appendixes

Glossary A

A

A-to-A

Refer to Application-to-Application.

Address Resolution Protocol (ARP)

A term used for routing on a LAN. ARP is used to map IP addresses into Ethernet addresses. ARP is not required for connection to ARPANET or MILNET, but is useful in the LAN workstation environment.

Alarm

A log message that is routed to an operator. Any CDCNET log message may be designated as an alarm.

Alarm History

A chronological record of the alarms received at a network operator's alarm buffer since the start of an operations session. An alarm history may be displayed using a network operations command.

Application-to-Application (A-to-A)

Can refer to either a type of link between two OSI layers, or a type of network processing:

1. An application-to-application link is an end-to-end link between an application layer of one system and the application layer of another for the exchange of information.

2. Application-to-application network processing that enables data to be exchanged between applications programs executing on different host computers or workstations.

ARP

Refer to Address Resolution Protocol.

ARPANET

A Defense Data Network (DDN) developed by the Defense Advanced Research Projects Agency. ARPANET supports research and development projects funded by the Department of Defense.

Asynchronous TIP

The terminal interface program (TIP) that configures terminal devices and establishes terminal attributes for a generic, asynchronous terminal connected to a device interface. The asynchronous TIP resides in a device interface that is configured to support asynchronous terminals.

Auto-Configured I/O Station

An I/O station that is logically configured and ready to use when the lines to which the devices in the I/O station become active and when a station operator connects to batch services. Contrast with Operator-configured I/O station. Configuring an auto-configured I/O station is possible when all the devices of the I/O station always connect to the same DI ports. Also known as a predefined I/O station.

B

Batch Device

Individual devices in an I/O station controlled by batch services and protocols and used for batch input and/or output. Examples of batch devices include card readers, line printers, card punches and plotters.

Block

In the context of network communications, a portion or all of a message. A message is divided into blocks to facilitate buffering, transmission, error detection, and correction for variable-length datastreams. Differing block protocols apply to the host-to-device interface and the device-interface-to-terminal interfaces.

During input from a terminal, a block is a single transmission consisting of one or more lines of one or more messages.

During input to a service, a block is a single line consisting of part or all of a message. Terminal transmission blocks are divided into as many service input blocks as needed, until the message is completed.

During output from a host application program, a block is one or more lines. During output from a device interface to a terminal, a block is one terminal transmission buffer.

Board

Refer to Logic Board.

Break 2 Sequence

A series of interactive terminal keystrokes that cause interruption in the datastream, stopping delivery of a message or output from the host. Some terminals are equipped with a single key that causes a break 2 sequence. Refer to your terminal user's manual for your terminal's exact sequence.

BSC3270 TIP

A terminal interface program that provides support for the IBM 3270 Information Display System. The 3270 Bisynchronous TIP allows 3271, 3274, 3275, and 3276 control units to connect directly to CDCNET in order to communicate with a CDCNET terminal device interface (TDI) over dedicated or dial-up lines using the centralized multipoint Binary Synchronous Communication protocol. The 3270 TIP Bisynchronous supports up to 32 multi-dropped clusters of up to 32 devices on each line.

Buffer

One of two structures for the storage of data in device interface memory. See also Data Buffer and Descriptor Buffer.

Byte

1. (ISO) A binary character string operated upon as a unit and usually shorter than a computer word.

2. (ISO) A group of contiguous bits. Unless prefixed (for example, a 6-bit byte), the term implies 8-bit groups. An 8-bit byte is sometimes called an octet. When used for encoding character data, a byte represents a single character.

C

Catenet

A group of connected CDCNET network solutions. This term is often used when referring to all the device interfaces and network solutions in a site's network.

Central Processor Unit (CPU)

The high-speed arithmetic processing unit that carries out the basic instructions required in program execution.

Channel

The physical link or logical path between a Mainframe Device Interface (MDI) and the network host computer, or between an Integrated Communication Adapter (ICA) and the Integrated Controller Interface (ICI) in the network host computer.

Clock Synchronization

A function that ensures that all device interfaces in a catenet are synchronized within 1 second of each other. Clock synchronization involves setting or resetting the master clock for the catenet (controlled by the Independent Clock ME) and synchronizing all of the device interface clocks in the catenet (controlled by the Dependent Clock ME in each device interface) according to the master clock. The DEFINE_SYSTEM command defines whether or not a device interface contains the Independent Clock ME.

On NOS, the device interface that contains the Independent Clock ME contains the master clock for the catenet, which synchronizes the rest of the clocks in the network.

On NOS/VE the Independent Clock ME is configured on the host.

Cluster Address

A sequence of bits, characters, or group of characters that identifies the location of a device (controller) that handles the remote communication processing for multiple (usually dumb) terminals or workstations.

Coaxial Cable

A transmission cable that provides large bandwidth and high data/low error rates. This cable contains a central carrier wire surrounded by fine copper mesh and/or an aluminum sleeve.

Code Set Procedure

A CDCNET load procedure that allows a site to define its own code sets for input and output devices.

Command File

A NOS file of network operations commands. Commands in the command file can be executed using the EXECUTE_COMMAND_FILE. Similar to a procedure file.

Communication Line

A terminal line that establishes a complete communication circuit between a terminal or workstation and a CDCNET device interface.

Configuration

The process by which various computer-related resources are coordinated to function together. Under CDCNET, various types of configuration activities are performed.

1. Network configuration, whereby hosts, terminals, workstations, and unit record devices are interconnected into a network using CDCNET device interfaces and appropriate communications media.

2. Device interface hardware configurations, whereby decisions are made regarding which logic boards to install in a particular CDCNET device interface.

3. Device interface software configuration, whereby CYBER hosts decide which CDCNET software to downline-load into a specific CDCNET device interface.

4. Creation of device interface configuration files, whereby network administrators or communications consultants identify/describe the specific CDCNET device interfaces that reside in their networks and place this information in host-maintained permanent files.

See also Logical Configuration.

Configuration Command

A command that establishes, cancels, or redefines the configuration of a network component in the network's logical definition.

Configuration File

Refer to Configuration Procedure.

Configuration Procedure

A procedure containing configuration commands that configure the software in a device interface. Each device interface has a unique configuration file, which is read whenever the device interface is reset and loaded. Also known as configuration file.

Configure

To define the variable attributes of a CDCNET device (such as the device interface, a single board, network solution, communication line or gateway). Examples of configurable attributes include buffer sizes, line speeds, and logical names.

Congested

One of the operational states of a network solution or communication line; indicates excessive traffic. See also Congestion.

Congestion

A condition in which there is more message traffic on a network solution or communication line than the line's carrying capacity. Continued congestion results in lengthy message delay and discarding of new messages.

Connection-Oriented Network Service (CONS)

OSI connection oriented network service specification used in conjunction with X.25.

CONS

See Connection-Oriented Network Service.

Control Facility

A NOS/VE service that monitors I/O stations and their batch devices, executes device and file control commands for the I/O station, and controls selection of files for output devices for the I/O station.

Cost

A relative measure assigned to a path (such as a network solution) that is used for transmitting data through a CDCNET-type network. The cost of each possible path is computed and stored into tables by the Routing Management Entity (ME). From these tables, the Routing ME determines the path that has the least cost. The path with the least cost is used to transmit data. The cost of a path may change depending upon the amount of congestion on the path. A congested network solution has a higher cost than an uncongested network solution.

Coupler

A hardware module on a Mainframe Device Interface (MDI) that connects a host's peripheral processor to CDCNET.

Coupler Node

A logical identification assigned to the coupler that connects a host channel and an MDI.

CPU

Refer to Central Processor Unit.

D

Data Buffer

A structure for storing user data in device interface memory. A pointer is associated with the first character of data in the buffer. Data buffer length is configurable. Contrast with Descriptor Buffer.

Datagram

A self-contained package of data carrying enough information to be routed from source to destination without reliance on earlier exchanges between source or destination and the transporting network.

DDN

Refer to Defense Data Network.

Deadman Timeout (DMTO)

A device interface hardware reset that occurs automatically if software does not work normally for 10 seconds.

Dedicated Line

A communication line that permanently connects a terminal to a device interface. Contrast with Switched Line.

Default

A pre-selected value supplied for a missing parameter upon the entry of a command or subcommand.

Defense Data Network (DDN)

A packet-switching network provided by the Department of Defense (DOD) to meet its current and projected data communication requirements. It is based upon the Defense Advanced Research Projects Agency Network (ARPANET), an existing operational network.

Descriptor Buffer

A data structure used for chaining data buffers. Contrast with Data Buffer.

DI Name Resolver

A program that resides in a DI and provides an interface between the DI and domain name servers. If a TCP/IP user specifies a domain name, the name resolver requests a domain name server to translate the name into the corresponding IP address.

Diagnostic

1. Software and/or microcode that isolates failing hardware/software components within a CDCNET device interface.

2. A message indicating a malfunction within a CDCNET device interface or one of its related communications media.

Dial-up Line

A communications circuit created by dialing a destination over a common carrier's switched lines.

Disabled

Cannot be used for normal network operation. Applies to boards, communication lines and network solutions.

DMTO

Refer to Deadman Timeout.

DOD

Department of Defense.

Domain Label

Part of a domain name and contains up to 63 characters. The label must begin with a letter (A..Z or a..z), which can be followed by letters, digits, or hyphens. The label must end with a letter or a digit.

Domain Name

TCP/IP users typically use domain names instead of IP addresses to reference TCP/IP services. Domain names identify hosts or other resources connected to a TCP/IP network. A domain name consists of a sequence of domain labels, arranged in a hierarchical order, and separated by periods. For example, the name PINK.ARH.CDC.COM, could specify a machine called PINK at the Arden Hills (ARH) facility of Control Data, which is a commercial organization (COM). The length of a domain name including the separating periods can be up to 255 characters.

Domain Name Server

A program that resides in a host connected to the TCP/IP network and responds to queries for information about domain names.

Down

A status of suspended service.

Dump

Refer to Memory Dump.

Dump Analyzer

CDCNET troubleshooting software that enables communications support analysts to review detailed memory dumps generated by malfunctioning CDCNET device interfaces. Refer to Analyze_CDCNET_Dump (ANACD).

E

Echoplex

A procedure in which the receiving station automatically retransmits each character received so that the sender may verify the correctness of his transmission. This process usually occurs on asynchronous full-duplex communication lines; however, not all terminals on full-duplex communication lines are capable of echoplex operation.

EEPROM

Refer to Electronically Erasable Programmable Read Only Memory.

EGP

Refer to Exterior Gateway Protocol.

Electronically Erasable Programmable Read Only Memory (EEPROM)

Read only memory that can be updated dynamically by the software at configuration time.

ESCI

Refer to Ethernet Serial Channel Interface.

Ethernet

A baseband local area network protocol developed by the Xerox Corporation. CDCNET supports an Ethernet-compatible network.

Ethernet Serial Channel Interface (ESCI)

The logic board within a CDCNET device interface that controls transmissions between an Ethernet (IEEE 802.3) transceiver and the internal system bus (ISB) of the device interface.

Exception List

A file that determines how to process the load requests of the network's device interfaces (DIs). The exception list is a file of commands that specify the version of software to be loaded into the device interface, and which error codes should trigger a dump of the device interface memory. There is one exception list for the network, containing a default entry and any exceptions to the default entry.

Exterior Gateway Protocol (EGP)

A TCP/IP protocol that allows for transfer and negotiation of routing information.

F

File Prefix Procedure (FPP)

A device configuration procedures containing strings of characters and/or control codes to be sent to the printer before every print file. An FPP is used for printers that use the PostScript language, such as the Apple LaserWriter.

File Transfer Protocol (FTP)

1. The Control Data application-to-application protocol that enables applications programs executing on a NOS or NOS/VE host to exchange information with applications programs that execute on other NOS or NOS/VE hosts.

2. TCP/IP protocol that provides the file transfer server and user functions.

Format Effectors

Any character used to control the positioning of printed or displayed data.

Forms Code

A 1- through 6-character identifier associating a print file with a certain printer form ensures output will be routed to a printer which prints in the format needed. For example, one printer at a site can be defined as using an 8-1/2 by 11-inch print form by specifying a forms code of DOC (document) on the command that configures the printer (DEFINE_BATCH_DEVICE). Another printer can be defined to print perforated checks and have a forms code of CHECKS, and one defined to print on carbon paper could have a forms code of CARBON. When output is routed to printers, the appropriate forms code (DOC, CHECKS, or CARBON) can be specified so that output will be printed by the appropriate printer.

FPP

Refer to File Prefix Procedure.

FTP

Refer to File Transfer Protocol.

G

Gateway

A software interface between systems with different architectures and protocols.

Gateway Title

The logical title assigned to a gateway during logical configuration.

H

HASP

Refer to Houston Automatic Spooling Program.

HASP Protocol

A job control protocol for transmitting data processing files and jobs between certain models of computers. It is also called the Houston Automatic Spooling Program.

HASP Workstation

A bisynchronous terminal with associated batch devices. HASP workstations are used for remote batch input from card readers and output to line printers, card punches, and plotters. Each workstation must have a console device that can be used as a normal interactive device with limited screen and formatting capabilities. Each HASP workstation can support the following: up to seven card readers; a combined total of eight batch output devices, (line printers and card punches which can be replaced with plotters), with a maximum of seven devices of the same type; and one console device (required).

HDLC

Refer to High-Level Data Link Control.

High-Level Data Link Control (HDLC)

The International Standards Organization's (ISO) bit-oriented protocol for the data link layer of the Open Systems Interconnection (OSI) reference model.

Hop

Within a network of interconnected gateways, a hop is the process of forwarding a packet from one gateway to another.

Host

Refer to Host Computer.

Host Computer

A mainframe computer system, connected to a communications network, which provides primary services, such as database access, user application execution, or program compilation. For CDCNET, a host computer provides network support functions, including maintenance of device interface load files. Also called a host.

Host Console

The keyboard and display screen used to manage the host computer. Also used in CDCNET to access the Network Operator Utility (NETOU) to monitor and control the CDCNET. See also System Console.

Host Operating System

The host containing applications and maintenance software available to the device interface.

Host Service Name

A logical name for the host computer. The host service name is the name that terminal users provide when connecting to the host using the CREATE_CONNECTION command.

Host System

A mainframe computer and its operating system that provides applications and services to the computer network. CDCNET must have at least one host running NOS, NOS/VE, or dual-state NOS and NOS/VE.

Houston Automatic Spooling Program (HASP)

A job control protocol for transmitting data processing files and jobs between certain models of computers.

I

I/O Station

A logical grouping of batch devices into a single named unit for routing jobs and files to the batch devices and for controlling the devices. Devices belonging to an I/O station may all connect to the same line, to several lines on one device interface, or to lines distributed among several device interfaces.

ICA

Refer to Integrated Communications Adapter.

Independent Log Management Entity (Independent Log ME)

1. Also known as the recorder logging function. Software resident in a host-connected device interface that works with the Independent File Access ME to write log messages generated by network device interfaces to a file on a host called the network log file.

2. A service on NOS/VE host computers that writes log messages generated by network device interface to a host-resident file called the network log file.

Initialization Procedure

A CDCNET load procedure that defines data to be sent to a printer when the printer's communication line becomes active.

Integrated Communications Adapter (ICA)

A hardware device that interconnects a single 16-bit Integrated Controller Interface (ICI) channel of a host computer with CDCNET. The ICA is installed in the CYBER 930 series host computer mainframe.

International Standards Organization (ISO)

A worldwide standards group similar in function to the American National Standards Institute (ANSI). ANSI is a member of International Standards Organization.

Internet Protocol (IP)

A term used in DDN networks that refers to a connectionless, point-to-point protocol corresponding to the CDCNET Internet layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

IP

Refer to Internet Protocol.

IP Address

Internet Protocol (IP) uses a 32-bit IP address field containing the Internet Address. Each IP system or host in the DDN is assigned a unique IP address. A host may have one or more IP addresses; however, a CDC CYBER host basically supports only one IP address per host.

ISO

Refer to International Standards Organization.

Isolation

Identification of a failing hardware or software component.

K

K Display

A NOS host console display that enables operators to interact with various operating system utilities (for example, those controlling user validation and NAM subsystem interaction).

L

LCA EEPROM

Refer to Logic Cell Array Electronically Erasable Programmable Read Only Memory.

Line

A circuit that connects a terminal to a device interface. A line is dedicated to carrying data to and from that terminal. It does not carry data that is routed through the rest of the network, nor does it use the CDNA protocol. Also known as a communication line.

LLC2

See Logical Link Control 2.

Load Procedure (LP)

A file containing commands specifying information to be downloaded to a printer. Load procedure types include initialization procedures (IPs), file prefix procedures (FPPs), code set procedures, and VFU load procedures (VLPs).

Log File

A file that is created and maintained by the operating system for storing error information and usage data concerning network elements.

Log Group

A logging function that is distributed among several device interfaces. A collection of device interfaces and the set of log messages associated with these device interfaces.

Log Management Entity (Log ME)

Software that manages the transmission and recording of log messages generated by device interface software. Consists of Dependent and Independent Log Management Entities. Dependent Log Management Entities, residing in device interfaces, are sources of log messages. Independent Log Management Entities, residing in a host-connected device interface, work with host applications or a NOS/VE host to write the log messages to the network's log file on the host.

Log Support Application (LSA)

Also known as the Dependent Log Management Entity and/or source logging function. Software that manages the generation and transmission of log messages generated by device interface software. Resident in every device interface.

Logging

The process of issuing messages for network activity and recording the messages in a log file.

Logic Board

A printed circuit board with data storage and/or processing components installed; sometimes called a board, card, or module.

Logic Cell Array Electronically Erasable Programmable Read Only Memory (LCA EEPROM)

Contains the configuration data for the XILINX logic cell arrays which contain the bulk of the random logic on the MPB-II. See also EEPROM.

Logical Configuration

The process of assigning names and values and setting variables throughout the CDCNET to define network elements (mainframes, terminals, lines, network solutions, device interfaces, gateways, and other elements), so that all network elements follow a uniform naming and addressing scheme. After logical configuration, network elements accept all data and commands directed to or through themselves, and reject all other data and commands. Also known as network definition.

Logical Link Control 2 (LLC2)

OSI connection-oriented data link protocol utilized by CONS/X.25 when running over ESCI.

Logical Name

A name assigned to a CDCNET component (device interface, network solution, communication line, gateway) in the logical definition of the network. Many network operations commands refer to CDCNET components by their logical names. Contrast with Title.

Logical Unit (LU)

A 3270 terminal device from which a 3270 terminal interface program (TIP) accepts an interactive session.

Loopback Test

A failure management test that checks the integrity of a hardware element by sending data through the element and back again.

LP

Refer to Load Procedure.

LSA

Refer to Log Support Application.

LU

Refer to Logical Unit.

M

Main Processor Board II (MPB-II)

Processor board containing a high performance architecture consisting of MC68030 32-bit processor and 512 K bytes of local onboard memory.

Main Processor Board (MPB)

The logic board within a CDCNET device interface that provides the primary processing power for the device interface.

Mainframe Channel Interface (MCI)

An optional logic board within a CDCNET device interface that connects the device interface to a 12-bit CYBER host channel.

Mainframe Device Interface (MDI)

The CDCNET device interface variant that interconnects a 12-bit channel of host computers operating under NOS or NOS/VE with an Ethernet (IEEE 802.3) local area network.

Mainframe/Terminal Device Interface (MTI)

The CDCNET device interface variant that interconnects 12-bit NOS and NOS/VE host computers with terminals, workstations, and unit record equipment without requiring a local area network.

Manage CDCNET Configuration (MANCC) Utility

A CDCNET host utility for NOS that helps create, edit, and display CDCNET configuration files.

Management Entity (ME)

CDCNET software that performs network management functions. CDCNET supports various MEs to perform specific network tasks.

MANCC

Refer to Manage CDCNET Configuration Utility.

MCI

Refer to Mainframe Channel Interface.

MDI

Refer to Mainframe Device Interface.

ME

Refer to Management Entity.

Memory Dump

The process and result of writing device interface memory to a host-resident file. Memory dumps are forced when the contents of device interface memory are at risk of being lost.

Metrics

Statistics which are collected and reported for CDCNET hardware and software components.

MILNET

A Defense Data Network (DDN) evolved from ARPANET that supports operational communication requirements.

Mode 4

A data communications protocol, consisting of variants 4A, 4B, and 4C. The Mode 4 protocol supports two-way alternate communications (where messages may be sent in one direction or another, but not in both directions simultaneously) on switched or dedicated synchronous lines within a line speed range of 1200 to 19200 bits-per-second.

The CDCNET Mode 4 terminal interface program supports the 4A and 4C variants of the Mode 4 protocol.

MPB

Refer to Main Processor Board.

MPB-II

Refer to Main Processor Board II.

MTI

Refer to Mainframe/Terminal Device Interface.

N

NAM

Refer to Network Access Method.

NAM K Display

A display on the host console screen that allows operator interface to Network Access Method (NAM). A CDCNET operator at the host console communicates with the CDCNET through the NAM K display.

NAM/VE

Refer to Network Access Method/Virtual Environment.

NDI

Refer to Network Device Interface.

NETCU

Refer to Network Configuration Utility.

NETLS

Refer to Network Log Server.

NETOPS

A NOS user name under which files are stored for use during CDCNET installation and by CDCNET-host operations. NETOPS contains files created and written by NAM while NAM is operating, the network directory file (NETDIR), and the NAMSTRT procedure.

NETOU

Refer to Network Operator Utility.

Network Access Method (NAM)

The access method that resides under NOS; allows host-based network applications programs to exchange information with communications networks.

Network Access Method/Virtual Environment (NAM/VE)

The access method that resides under NOS/VE; allows host-based network applications programs to exchange information with communications networks.

Network Architecture

A set of functional layers in which each layer performs a specific set of functions and services; together, the layers interact to provide total, end-to-end network operation. Each layer uses a protocol and has its relationship with other layers defined.

Network Configuration Utility (NETCU)

A CDCNET utility on NOS/VE that logically configures CDCNET.

Network Definition

The process of assigning logical names to network components and assigning values to variable parameters for CDCNET software. See also Logical Configuration.

Network Delay Measurement

A software feature used to measure one-way delay time between two network DIs.

Network Device Interface (NDI)

The standard CDCNET device interface variant that transfers data between networks (for example, between two local area networks; between a local area network and a communications line; or between a local area network and a public data network).

Network File System (NFS)

A software product of Sun Microsystems, Inc. that allows a variety of machines and operating systems to share files.

Network Identifier

A unique identifier (32-bit character string) assigned to a network solution.

Network Job Entry Facility (NJEF)

The network applications software that supports IBM's Network Job Entry (NJE) protocol on NOS.

Network Log File

A file on a host computer that contains CDCNET log messages sent from the network's device interfaces and serves as a record of the network's activity.

Network Log Server (NETLS)

A CDCNET host application that writes CDCNET log messages generated by device interfaces to the network log file on the host.

Network Logfile Termination (NLTERM) Utility

A CDCNET host utility on NOS that terminates the currently-active network log file to which NETLS is writing log messages, and renames the terminated log file. NLTERM also provides information about previously-terminated log files as an aid in managing log files.

Network Operator

A person who monitors CDCNET activity, has the ability to control CDCNET hardware and software, makes occasional network configuration changes, and performs elementary troubleshooting by sending commands to the network's device interfaces. A network operator may perform these tasks from a host console or a remote terminal.

Network Operator Utility (NETOU)

A group of programs residing on a host computer and in a (NOS) mainframe device interface or mainframe terminal interface connected to the mainframe. NETOU allows a network operator to access, monitor, control, and configure a CDCNET from the host console or a remote terminal. Using NETOU, network operators can send CDCNET operations commands to specific device interfaces or to all the device interfaces in the network.

Network Performance Analyzer (NPA)

The CDCNET software utility that generates statistical reports based on its analysis of the network log file or generates event/error reports based on log messages in the network log file.

Network Products Gateway

A gateway that allows information transfer between CDCNET and a non-CDNA host such as a NOS host. File transfers between NOS hosts over CDCNET require Network Products gateways to be defined in the MDIs connected to the hosts.

Network Products (NP)

Programs that run under NOS in a host mainframe to allow data and computer applications to be transmitted from the mainframe through a computer network. Network Products include Network Access Method (NAM) and Network Definition Language (NDL). Network Products and CDCNET have different architectures. For hosts to send data through CDCNET, the Mainframe Device Interfaces connected to the mainframes must have gateways to translate between Network Products and CDCNET protocols.

Network Products Terminal Gateway

A gateway that allows both interactive and remote batch terminal users to connect to a NOS host through CDCNET (by specifying the appropriate service title on the CREATE_CONNECTION command). There are two parts to the NP Terminal gateway: the Interactive Virtual Terminal gateway (IVT gateway) and the Remote Batch Facility gateway (RBF gateway). The batch gateway is dependent on the interactive gateway. If a network configuration is going to support terminal connections to NOS, the MDI or MTI connected to the NOS host must contain an NP Terminal gateway.

Network Service Access Point Address (NSAP Address)

An address used in the OSI protocol stack that uniquely identifies a CDCNET system and a user of the OSI Network layer within that system. An NSAP address consists of two parts: a Network Entity Title and an NSAP selector. The Network Entity Title uniquely identifies a CDCNET system. The NSAP selector uniquely identifies a user of the OSI Network layer in that system.

Network Solution

A communications medium over which data is transmitted between interconnected network resources, and which uses CDCNET protocols. In OSI terminology, a network solution is also referred to as a subnet. A network solution differs from other communications lines because it is shared by multiple network resources (it is not solely dedicated to the handling of data transmissions between a single pair of network resources). Network solutions differ from trunks because they can carry network management traffic such as log and alarm messages.

Network Transfer Facility (NTF)

An application providing a fully symmetric queue file transport facility between a NOS/VE host and another host in a geographically dispersed network. NTF supports IBM's Network Job Entry (NJE) protocol and HASP multileaving protocol for communication between hosts.

Network Validation

A system security feature requiring users to enter a valid username and password to use CDCNET.

NFS

Refer to Network File System.

NJEF

Refer to Network Job Entry Facility.

NLTERM

Refer to Network Logfile Termination Utility (NLTERM).

NP

Refer to Network Products.

NP IVT Gateway

Network Products Interactive Virtual Terminal Gateway. A program which runs in a Mainframe Device Interface (MDI) or Mainframe Terminal Device Interface (MTI) connected to a host mainframe, and which allows the host mainframe to send applications through CDCNET to interactive terminals. The gateway acts as a protocol converter between the host's Network Products protocols and CDCNET protocols. Also known as the NP terminal gateway.

NP Terminal Gateway

Refer to NP IVT Gateway.

NPA

Refer to Network Performance Analyzer.

NSAP Address

Refer to Network Service Access Point Address.

NTF

Refer to Network Transfer Facility.

NVT

Refer to TELNET Network Virtual Terminal.

O

Octet

An 8-bit byte.

Online Diagnostics

Optional diagnostics for the device interface that can be executed while the device interface is connected to and operating as part of the CDCNET.

Online Loader

A CDCNET service that loads software into device interfaces when the software is needed while the network is operational, as opposed to initial loader, which loads software into device interfaces only when they are started up (initialized).

Open System Interconnection (OSI)

The International Standards Organization's (ISO's) reference model for network processing. This model is based on a network architecture that segregates network functions into seven layers.

Operations Station

The remote terminal or host console from which CDCNET network operations are performed through the Network Operations Utility (NETOU).

Operator-Configured I/O Station

An I/O station that is logically configured when an I/O station operator invokes a terminal definition procedure (TDP) to define the I/O station. The station operator must define the I/O station before it can be used, and the devices in the I/O station are not active until the TDP executes. Contrast with Auto-configured I/O Station. Configuring an Operator-configured I/O station is necessary when the devices of an I/O station do not always connect to the same device interface port. An example of an Operator-configured I/O station is a dial-up HASP workstation. Also known as a dynamically defined I/O station.

Operator Console

An interactive terminal in an I/O station that can be used to control the other batch devices in the I/O station. On NOS/VE, the operator console is used for entering OPERATE_STATION (OPES) utility subcommands to control the devices. On NOS, the operator console is used for entering Remote Batch Facility (RBF) commands to control the devices.

OSI

Refer to Open System Interconnection.

Outcall Gateway

A gateway that provides both terminal and device outcall services. See also Gateway.

P

Packet Assembly/Disassembly (PAD)

(ISO) A functional unit that enables data terminal equipments (DTEs) not equipped for packet switching to access a packet-switched network.

PAD

Refer to Packet Assembly/Disassembly.

Page Memory Management Unit (PMMU)

Provides address translation and memory protection for a demand paged virtual memory system.

Passthrough

Refer to Terminal Passthrough.

PDN

Refer to Public Data Network.

Physical Name

A name assigned to a hardware device in a device interface: boards, ports, and memory banks, such as $CIM3 (physical name for CIM board in slot 3) and $LIM5_PORT2 (physical name for second port on LIM board in slot 5.)

Physical Record Unit (PRU)

The amount of information transmitted by a single physical operation of a specified device. For mass storage files, a PRU is 64 central memory words (640 characters); for magnetic tape files, the size of the PRU depends upon the tape format. A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

PMM

Refer to Private Memory Module.

PMMU

Refer to Page Memory Management Unit.

Port

The physical connection on the device interface through which data is transferred to/from the device interface. Each port is numbered and supports a single communication line.

PostScript

An industry standard page description language for describing text, graphic entities, and digitized images for printed pages. PostScript can also be used to control aspects of a printer's operation. PostScript page descriptions are programs run by an interpreter in the printer. PostScript programs are generated by application programs running on a system to which the printer is connected.

Primary MDI

The Mainframe Device Interface (MDI) to which the operator sends commands and receives responses and alarms. At any time, only one MDI can communicate with the operator.

Printer Support Utility (PSU)

The network applications software that supports standalone CDCNET printers on NOS.

Private I/O Station

An I/O station used to submit and receive jobs and output files only for the user that is operating it. A station operator must monitor and control the I/O station for it to be active. Contrast with Public I/O Station.

Private Memory Module (PMM)

The logic board within a CDCNET device interface that provides additional random access memory dedicated for use by the main processor board (MPB) of the device interface.

Program EEPROM

Refer to Program Electronically Erasable Programmable Read Only Memory.

Program Electronically Erasable Programmable Read Only Memory (Program EEPROM)

Contains the boot and diagnostic code for the MPB-II. Also contains the PMMU translation tables. Mostly synonymous with MPB ROM on the MPB-I. See also EEPROM.

Programming System Report (PSR)

An official report to Control Data of a problem with Control Data software. A PSR can be sent to Control Data either in hard-copy form, or by using the on-line SOLVER program.

Protocol

A set of conventions that must be followed to achieve complete communications between the computer-related resources in a network. A protocol can reflect the following:

1. A set of pre-defined coding sequences, such as the control byte envelopes added to (or removed from) data exchanged with a terminal.

2. A set of data addressing and division methods, such as the block mechanism used between a network application program and Network Access Method.

3. A set of procedures that control communications, such as the supervisory message sequences used between a network application program and Network Access Method.

Protocol Stack

A collection of protocols in successive layers. CDCNET is based on ISO's Open System Interconnection (OSI) reference model, where each system includes a set of layers and each layer supports one or more protocols.

CDCNET phase 2 of OSI support includes support for OSI and TCP/IP protocols for layers 3 and 4. Therefore, CDCNET supports two protocol stacks: OSI and TCP/IP.

PRU

Refer to Physical Record Unit.

PSR

Refer to Programming System Report.

PSU

Refer to Printer Support Utility.

Public Data Network (PDN)

A commercial packet-switching network that supports the communications interface described in CCITT protocol X.25.

Public I/O Station

An I/O station shared by many users who may submit jobs through it and receive output. The operator who controls a public I/O station does not own the files sent to or read from it. Routing of output files for a public I/O station is controlled through the I/O station's name. A station operator does not have to monitor and control a public I/O station for it to be active. Contrast with Private I/O Station.

PVC

Permanent virtual circuit.

R

Radix

The base of a number system. For example, 2 is the binary system radix and 10 is the decimal system radix.

RBF

Refer to Remote Batch Facility.

Recorder Log Group

A logging function in which device interfaces that are sources of log messages report their log messages to a device interface which works with a host application to record the log messages in a network log file. The Independent Log ME controls the log recording function.

Relay

Process occurring when CDCNET receives a data unit from a directly connected network solution and transmits the data unit to another directly connected network solution.

Remote Batch Facility (RBF)

The network applications software that supports remote batch processing (remote job entry) on NOS.

Remote Line Monitor

Displays and/or records all received and transmitted characters on an LIM and port supported by the standard CDCNET CIM firmware and that use protocols defined for Remote Line Monitor.

RS-232-C

An Electrical and Electronic Industries Association (EIA) standard that describes the interface between terminals or other Data Terminal Equipment (DTE) and modems or other Data Communications Equipment (DCE) employing a serial binary interchange.

RS-449

1. A physical interface standard for data communications used with high speeds and long communication lines.

2. A newer standard than RS-232-C, also used for serial communications. Eventually meant to replace RS-232-C, but backward compatibility is specified in RS-449.

S

SCL

Refer to System Command Language.

SCL Comment

A comment within a SCL command. The comment is enclosed by quotation marks and is ignored during command processing.

SDLC

Refer to Synchronous Data Link Control.

Segment

The unit of data exchanged by TCP modules. This term also describes the unit of exchange between any transport protocol modules.

Server TELNET

Provides a mechanism for an interactive terminal that uses TCP/IP TELNET services on a foreign host to communicate with the interactive services of NOS/VE.

Service

An entity that is external to CDCNET but is registered within CDCNET as being capable of conducting input and output with a terminal or with another service. Services have names. Terminal users connecting to a host are connecting to a service. An example of a service is the Interactive Facility (IAF) on a host.

Simple Mail Transfer Protocol (SMTP)

A mail exchange protocol used between hosts on a TCP/IP network. SMTP does not define the end-user interface. However, SMTP provides a program interface to the local mail system.

SMM

Refer to System Main Memory.

SMM4

A 4 M byte version of the SMM (see System Main Memory).

SMTP

Refer to Simple Mail Transfer Protocol.

SNA

Refer to Systems Network Architecture.

SNA3270 TIP

A terminal interface program that provides IBM 3270 Information Display System users access to CDCNET through an SNA network.

SNPA Address

Subnetwork point of attachment address. An address representing the attachment or connection of a system to a subnet. Generally, the SNPA address is a layer 1 address. For Ethernet, the Ethernet station ID represents the SNPA address. For an X.25 subnet, the DTE address represents the SNPA address.

Socket

A TCP/IP address used to locate a process on a host. This address is used by Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). It consists of the 32-bit IP address and a 16-bit port number.

SOLVER

An online utility maintained by Control Data that contains a database of reported software problems and solutions. SOLVER can be used for writing a PSR to report a problem with software.

Source Log Group

A logging function in which device interfaces that are sources of log messages maintain a list of log messages which they will send to recorder device interfaces. The source logging function is controlled by the Dependent Log ME, also known as Log Support Application (LSA).

SRI International

A network information center (NIC) providing administrative support services to the Department of Defense for TCP/IP and DDN networks.

Station Operator

A person in charge of controlling batch devices in an I/O station by sending commands to the equipment from the station operator console. On NOS/VE, the station operator uses OPERATE_STATION (OPES) utility commands to control the devices. On NOS, the station operator uses the Remote Batch Facility (RBF) commands to control the devices.

Statistics

Refer to Metrics.

Status

Information about the current state of a network component: Device Interface (DI), the hardware components (boards, ports) of a device interface, lines and network solutions connected to the device interface, and device interface software.

Status Command

A command that requests and displays the operational status of a particular network component, such as a device interface or a network solution.

Subnet

The concept of a subnet is used both in OSI and TCP/IP. Therefore, there are two definitions of subnet.

1. OSI: In CDCNET, individual systems are connected to each other via different media such as Ethernet and HDLC. A medium connecting two or more CDCNET systems is called a network solution or a subnet. For OSI, a subnet refers to one and only one network solution.

2. TCP/IP: TCP/IP subnetting is a required Internet standard. Subnetting allows a configuration consisting of many physical networks to be addressed with a single IP network number. Each physical network is assigned a subnet number. Each host is addressed based on its network number, subnet number, and host number. These three fields make up the IP address.

Subnet Identifier

An identifier that identifies a subnet in a CDCNET network. It must be unique, and must not be associated with more than one subnet in a CDCNET network.

Switched Line

A communication line connected with one device interface, but able to be connected to any one of several terminals via a switching mechanism, such as a dialed telephone line. Contrast with Dedicated Line.

Synchronous Command Entry Mode

A command control mechanism that prevents operators from entering a command before a previously sent command has executed and returned a response.

Synchronous Data Link Control (SDLC)

Bit-oriented data link control protocol developed by International Business Machines (IBM).

System Address

The unique address assigned to a device interface in the network. The system address corresponds to the system title, so that commands and data sent by system title are received at the proper device interface address. See also System Identifier.

System Command Language (SCL)

The NOS/VE command language on which CDCNET network operations, and configuration and terminal user commands are based.

System Console

A component of a host operating system that is used to monitor and control the operating system. The system console can also be used to monitor and control CDCNET through the Network Operator Utility (NETOU). See also Host Console.

System Identifier

At the time of its manufacture, each device interface is assigned a unique 48-bit identification number from a pool of numbers allocated to Control Data by Xerox. This number is written into battery-backed RAM and is used throughout the catenet as the system identifier for that device interface.

The system identifier is used as the Ethernet address for any system that is locally connected to one or more Ethernet network solutions.

See also System Address.

System Main Memory (SMM)

A device interface board containing dynamic RAM accessible by all interfaces and the resident main processor board (MPB).

System Title

The title assigned to a device interface during logical configuration. This title corresponds to the device interface's system address, so that commands sent to a device interface by system are received at the proper device interface address.

Systems Network Architecture (SNA)

IBM standard defining the layers and layer protocols to be used within an IBM network.

SYSTEMX

A NOS user name that is used to store files for NOS and CDCNET installation and operations.

T

T-to-A

Refer to Terminal-to-Application.

TCP

Refer to Transmission Control Protocol.

TCP/IP

Refer to Transmission Control Protocol/Internet Protocol.

TDI

Refer to Terminal Device Interface.

TDP

Refer to Terminal Definition Procedure.

TELNET Network Virtual Terminal (NVT)

A TCP/IP protocol that provides presentation layer services for other application protocols. TELNET NVT protocol is roughly equivalent to VTP in the ISO model. It establishes connections and controls interactive virtual circuits.

Terminal Definition Procedure (TDP)

An optional configuration file that defines a terminal device or devices connected to a line whenever the line becomes active. A TDP can be used to define a terminal device that differs from the default terminal device type defined by the TIP that controls the line.

Terminal Device Interface (TDI)

The CDCNET device interface variant that interconnects terminals, workstations, and unit record devices with an Ethernet local area network.

Terminal Interface Program (TIP)

CDCNET software that resides in terminal device interfaces (TDIs) and enables terminals/workstations that employ specific terminal protocols (such as async, HASP, and IBM 3270) to communicate in CDCNET networks.

Terminal Passthrough

A CDCNET feature that allows interactive asynchronous terminal traffic to pass through the network transparently. The hosts and terminals interface with each other as if they were directly connected. Terminal passthrough allows a CDCNET-connected terminal user to access non-CDCNET supported hosts, such as NOS/BE and VAX.

Terminal-to-Application (T-to-A)

A type of network processing that enables the exchange of data between applications programs that reside on host computers and user terminals or workstations. In this case, protocol conversions occur so that transmitted data is understood both at the host and at the terminal or workstation.

Terminal User Procedure (TUP)

An optional configuration file that defines attributes of terminals and connections. A TUP can be used to define attributes for a particular terminal model or a group of terminals. A TUP for a terminal is executed when the communication line from the terminal to the supporting device interface becomes active.

Test

Software and/or microcode that provides detection and confidence capabilities. Also known as a diagnostic.

TIP

Refer to Terminal Interface Program.

Title

A string of 1 through 255 ASCII characters that identify a network service component such as a device interface or a gateway. The Directory Management Entity refers to the component by its title.

A name used to identify services available in the network. Titles are known throughout the catenet. Contrast with Logical Names, which are local to individual device interfaces.

Transmission Control Protocol/Internet Protocol (TCP/IP)

The name given to a suite of protocols that support the ARPANET community. TCP/IP protocol implementation is required within CDCNET for connectability to Defense Data Networks (MILNET or ARPANET) and to workstations that use TCP/IP.

Transmission Control Protocol (TCP)

A term used in DDN networks that refers to an end-to-end, connection-oriented protocol corresponding to the CDCNET Transport layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

Transmission Media

Provides the physical channel used to interconnect device interfaces in a network.

Transport Layer

Open Systems Interconnection (OSI) layer 4. Provides end-to-end control of a communication session once the path has been established. It allows processes to exchange data reliably and sequentially, regardless of which systems are communicating.

Trunk

A logical definition of a line and the communications software that allows the line to carry data between communications controllers. These controllers could be device interfaces or devices for other networks. Trunks going to other networks, such as DECNET or SNA, are not recognized as network solutions.

TUP

Refer to Terminal User Procedure.

U

UDP

See User Datagram Protocol.

ULP

Refer to Upper Layer Protocols.

Unit Record Interface (URI)

A Line Interface Module (LIM)-type peripheral circuit board that interfaces with the LIM bus and is used with the Communications Interface Module (CIM). The URI provides an 8-bit parallel interface for the operation of character or line printer. The URI includes all necessary drivers, receivers, timing, and control circuitry to drive one printer at a time.

Upper Layer Protocols (ULP)

A collective term for layers 5, 6, and 7 of the Open System Interconnection (OSI) network reference model.

URI

Refer to Unit Record Interface.

User Datagram Protocol (UDP)

A layer of TCP/IP interface software. UDP provides datagram-oriented services that are unreliable (connectionless), but low overhead, to upper layer protocols such as Domain Name resolver and server and NFS.

User TELNET

Allows a CDCNET terminal to connect to a foreign host's interactive service using TCP/IP TELNET communications.

V

VE Interface

A channel between a NOS/VE host and an MDI or ICA-II that uses the OSI protocol stack.

Version

A four-digit hexadecimal number indicating the release version of the software loaded in a device interface.

Vertical Format Unit (VFU) Load Image

A fixed or loadable image that defines format control channels and vertical spacing for a printer.

VFU

Refer to Vertical Format Unit (VFU) Load Image.

VFU Load Procedure (VLP)

A vertical format unit (VFU) load image that is defined in a procedure file. Commands in the procedure file specify the location of printer format control channels. When the procedure file executes, a binary version of the VFU load image is loaded into the printer.

Virtual Circuit

A connection between a source and a receiver in a network that may be realized by different circuit configurations during data transmission. Also called a logical circuit.

VLP

Refer to VFU Load Procedure.

W

WAN

Refer to Wide Area Network.

Well-Known Port

Ports used in TCP to name the ends of logical connections which carry long-term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. A contact port is sometimes referred to as a well-known port.

Wide Area Network (WAN)

A wide area network (WAN) such as ARPANET or DDN.

Wildcard Characters

Characters that can be used in place of other characters as variables. Wildcard characters can be used to replace single characters, to replace strings of characters, or to match characters to those specified in a list.

X

X.PC

An asynchronous data communications protocol that improves the networking capabilities of personal computers. It also allows users to have multiple active virtual circuits.

X.25 Asynchronous TIP

Also known as X.29 PAD, this is a CDCNET feature that allows asynchronous terminals to access CDCNET either by a Public Data Network (PDN) that supports the X.3 Packet Assembly/Dissassembly (PAD) facility or by the terminals operating in X.25 mode.

X.25 Gateway

A gateway used to transfer data from a host connected to CDCNET to a host in another network at the other end of the X.25 circuit. The X.25 gateway allows host-to-host (A-to-A) connections to take place over an X.25 circuit. A-to-A connections over X.25 circuits are provided by the Network Products applications.

XID

An identifier used for SNA3270 configurations. The XID is a psuedo-model ID that identifies a DI in the SNA network. CDCNET commands use the variable part of the XID, which contains five hexadecimal digits called the terminal identifier. An SDLC station returns this identifier in an SDLC exchange identification command. The DI adds a fixed prefix to this variable part to create a 6-byte XID.

Character Set B

This appendix lists the ASCII character set, with conversions to decimal, hexadecimal and octal codes.

Table B-1. ASCII Character Set

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
000	00	000	NUL	Null
001	01	001	SOH	Start of heading
002	02	002	STX	Start of text
003	03	003	ETX	End of text
004	04	004	EOT	End of transmission
005	05	005	ENQ	Enquiry
006	06	006	ACK	Acknowledge
007	07	007	BEL	Bell
008	08	010	BS	Backspace
009	09	011	HT	Horizontal tabulation
010	0A	012	LF	Line feed
011	0B	013	VT	Vertical tabulation
012	0C	014	FF	Form feed
013	0D	015	CR	Carriage return
014	0E	016	SO	Shift out
015	0F	017	SI	Shift in
016	10	020	DLE	Data link escape
017	11	021	DC1	Device control 1 (X-ON)
018	12	022	DC2	Device control 2
019	13	023	DC3	Device control 3 (X-OFF)
020	14	024	DC4	Device control 4
021	15	025	NAK	Negative acknowledge
022	16	026	SYN	Synchronous idle
023	17	027	ETB	End of transmission block
024	18	030	CAN	Cancel
025	19	031	EM	End of medium
026	1A	032	SUB	Substitute
027	1B	033	ESC	Escape
028	1C	034	FS	File separator
029	1D	035	GS	Group separator
030	1E	036	RS	Record separator
031	1F	037	US	Unit separator

(Continued)

Table B-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
032	20	040	SP	Space
033	21	041	"	Exclamation point
034	22	042	#	Quotation marks
035	23	043		Number sign
036	24	044	$	Dollar sign
037	25	045	%	Percent sign
038	26	046	&	Ampersand
039	27	047	'	Apostrophe
040	28	050	(	Opening parenthesis
041	29	051	)	Closing parenthesis
042	2A	052	*	Asterisk
043	2B	053	+	Plus
044	2C	054	,	Comma
045	2D	055	-	Hyphen
046	2E	056	.	Period
047	2F	057	/	Slant
048	30	060	0	Zero
049	31	061	1	One
050	32	062	2	Two
051	33	063	3	Three
052	34	064	4	Four
053	35	065	5	Five
054	36	066	6	Six
055	37	067	7	Seven
056	38	070	8	Eight
057	39	071	9	Nine
058	3A	072	:	Colon
059	3B	073	;	Semicolon
060	3C	074	<	Less than
061	3D	075	=	Equals
062	3E	076	>	Greater than
063	3F	077	?	Question mark
064	40	100	@	Commercial at
065	41	101	A	Uppercase A
066	42	102	B	Uppercase B
067	43	103	C	Uppercase C
068	44	104	D	Uppercase D
069	45	105	E	Uppercase E
070	46	106	F	Uppercase F
071	47	107	G	Uppercase G

(Continued)

Table B-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
072	48	110	H	Uppercase H
073	49	111	I	Uppercase I
074	4A	112	J	Uppercase J
075	4B	113	K	Uppercase K
076	4C	114	L	Uppercase L
077	4D	115	M	Uppercase M
078	4E	116	N	Uppercase N
079	4F	117	O	Uppercase O
080	50	120	P	Uppercase P
081	51	121	Q	Uppercase Q
082	52	122	R	Uppercase R
083	53	123	S	Uppercase S
084	54	124	T	Uppercase T
085	55	125	U	Uppercase U
086	56	126	V	Uppercase V
087	57	127	W	Uppercase W
088	58	130	X	Uppercase X
089	59	131	Y	Uppercase Y
090	5A	132	Z	Uppercase Z
091	5B	133	[	Opening bracket
092	5C	134	\	Reverse slant
093	5D	135	]	Closing bracket
094	5E	136	^	Circumflex
095	5F	137	_	Underline
096	60	140		Grave accent
097	61	141	a	Lowercase a
098	62	142	b	Lowercase b
099	63	143	c	Lowercase c
100	64	144	d	Lowercase d
101	65	145	e	Lowercase e
102	66	146	f	Lowercase f
103	67	147	g	Lowercase g
104	68	150	h	Lowercase h
105	69	151	i	Lowercase i
106	6A	152	j	Lowercase j
107	6B	153	k	Lowercase k
108	6C	154	l	Lowercase l
109	6D	155	m	Lowercase m
110	6E	156	n	Lowercase n
111	6F	157	o	Lowercase o

(Continued)

Table B-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
112	70	160	p	Lowercase p
113	71	161	q	Lowercase q
114	72	162	r	Lowercase r
115	73	163	s	Lowercase s
116	74	164	t	Lowercase t
117	75	165	u	Lowercase u
118	76	166	v	Lowercase v
119	77	167	w	Lowercase w
120	78	170	x	Lowercase x
121	79	171	y	Lowercase y
122	7A	172	z	Lowercase z
123	7B	173	{	Opening brace
124	7C	174	|	Vertical line
125	7D	175	}	Closing brace
126	7E	176	~	Tilde
127	7F	177	DEL	Delete

DI Reset Codes C

This appendix lists the DI reset codes numerically and suggests actions based on them. Table C-1 provides the numerical list of reset codes. The Action Code column in table C-1 is keyed to the suggested actions, which follow the table.

Table C-1. Numerical List of DI Reset Codes

Numeric Code	Reason Code	Issuing Component	Action Code
00(16)	power_up_reset	MPB ROM	DA
02(16)	manual_reset	MPB ROM	DA
03(16)	halt_memory_fault	MPB ROM	HW
04(16)	dead_man_time_out	MPB ROM	SW
05(16)	pp_channel_master_clear	ICA Boot	DA
06(16)	reset_function	ICA Boot	DA
08(16)	sram_parity_error_reset	MPB-II ROM	HW
10(16)	load_software_too_big	Initialization Bootstrap	LF
11(16)	improper_first_module	Initialization Bootstrap	LF
12(16)	unsatisfied_external	Initial Loader	LF
13(16)	sysconfig_not_loaded	Initial Loader	LF
14(16)	post_load_routines_not_found	Initial Loader	LF
15(16)	reset_at_end_of_quiesce	Initialization Bootstrap	DA
16(16)	unrecognizable_object_text	Initial Loader	LF
17(16)	duplicate_entry_point	Initial Loader	LF
18(16)	task_error_no_recovery_proc	System Ancestor	SW
19(16)	task_error_exceed_max_recovers	System Ancestor	SW
1a(16)	task_error_unrecoverable	System Ancestor	SW
1b(16)	no_configuration_file_obtained	Configuration Procurer	OP
1c(16)	configuration_file_read_error	Configuration Procurer	OP
1d(16)	not_enough_memory_for_buffers	Loader	LF
1e(16)	identification_record_expected	Loader	LF
1f(16)	unexpected_idr_encountered	Loader	LF
20(16)	premature_eof_on_file	Loader	LF
21(16)	absolute_length_too_large	Loader	LF
22(16)	invalid_object_text_version	Loader	LF
23(16)	invalid_module_kind	Loader	LF
24(16)	invalid_module_attribute	Loader	LF
25(16)	invalid_section_ordinal	Loader	LF
26(16)	duplicate_section	Loader	LF
27(16)	invalid_section_kind	Loader	LF
28(16)	invalid_allocation_alignment	Loader	LF
29(16)	invalid_offset	Loader	LF
2a(16)	storage_allocation_failed	Loader	OP/LF
2b(16)	undefined_section	Loader	LF
2c(16)	reference_outside_of_section	Loader	LF
2d(16)	invalid_address_kind	Loader	LF
2e(16)	invalid_number_of_bytes_spanned	Loader	LF
2f(16)	transfer_sym_entry_pt_not_found	Loader	LF
30(16)	parameter_verification_error	Loader	LF
31(16)	loader_table_not_found	Loader	LF

(Continued)

Table C-1. Numerical List of DI Reset Codes *(Continued)*

Numeric Code	Reason Code	Issuing Component	Action Code
32(16)	kill_system_with_dump	KILS Command	DA
33(16)	kill_system_without_dump	KILS Command	DA
34(16)	stop_executive	Executive	SW
35(16)	module_checksum_is_invalid	System Audits	LF
36(16)	software_dead_stop	DEAD STOP	SW
37(16)	fatal_parity_error	Executive	HW
38(16)	ac_low_error	Executive	OP
39(16)	temperature_shutdown_error	Executive	OP
3A(16)	reset_from_debugger	Hardwired in Debugger	DA
3B(16)	overflowed_stack	Exec/System Audits	SW
3C(16)	system_data_not_found	Initial Loader	LF
3D(16)	boot_file_media_mismatch	Boot Start-up Code	OP/LF
3E(16)	cybil_detected_error	CYBIL Routines	SW
3F(16)	hard_failure	Executive	HW/SW
40(16)	well_known_configuration_change	Configuration Procurer	NA
41(16)	mpb_ram_ptr_not_found	Initial Loader	LF
42(16)	timer_task_module_missing	Initial Loader	LF
43(16)	task_received_unknown_itm	Any Task	SW
44(16)	sna-3270_tip_dhcf_abort	SNA 3270 TIP_DHCF	SW
45(16)	configuration_cmd_read_error	Configuration Procurer	OP
46(16)	eeprom_updated	Configuration Procurer	NA
47(16)	loader_bus_error	Initial Loader	LF

Suggested Actions Based on DI Reset Codes

The remainder of this appendix describes the circumstances in which DIs reset and suggests actions to be taken based on various DI reset codes. This information is keyed to the Action Code column in table C-1 through the abbreviation given for each reset title. Reset code descriptions are organized numerically within the action code groups.

Some of the actions suggested here require tools or facilities that might not be available at your site. If you need further assistance, submit a programming system report (PSR) to Control Data.

Deliberate Action (DA)

These resets are due to human intervention. For resets that generate dumps, the following steps can be taken to obtain more information:

- Display the executive error table with the Dump Analyzer DISEET subcommand.
- Use the DISSCT subcommand to check for memory and/or buffer regulation.
- Display calls for the running task.
- Use the DISC subcommand to find any task calling DEAD_STOP, RESET_DI, or ABORT_SYSTEM.

Following are descriptions of the causes and suggested actions for the DI resets classified as deliberate actions:

00(16) = POWER_UP_RESET

No dump file is written under this condition.

02(16) = MANUAL_RESET

The toggle switch on the MPB was manually reset. Additional information should be obtained from the person who reset the system.

05(16) = PP_CHANNEL_MASTER_CLEAR

The ICA-II is reset during the host deadstart.

06(16) = RESET_FUNCTION

The ICA-II is reset via a reset function from the PP.

15(16) = RESET_AT_END_OF_QUIESCE

This occurs if a DI is manually reset while the onboard diagnostics are running, or if there was a channel error. If the host error log indicates a channel error, follow the hardware error reporting process. See chapter 5.

32(16) = KILL_SYSTEM_WITH_DUMP

The system was reset by the KILL_SYSTEM operator command. Additional information should be obtained from the person who reset the system.

33(16) = KILL_SYSTEM_WITHOUT_DUMP

The system was reset by the KILL_SYSTEM operator command. Additional information should be obtained from the person who reset the system. No dump file is written under this condition.

3A(16) = RESET_FROM_DEBUGGER

The RS command was entered from the DI Resident Debugger. Additional information should be obtained from the person who reset the system.

No Action (NA)

This type of reset does not require any human intervention.

40(16) = WELL_KNOWN_CONFIGURATION_CHANGE

The system was reset to immediately and automatically force the changes specified in the configuration file for the MPB RAM.

There are a number of configurable values (such as which protocol stacks are enabled and data buffer size) that are associated with this reset. In addition, if a change in configuration affects the allocation of PMM, reset 40 is invoked. In any case, a dump is never taken when a reset 40 occurs.

46(16) = EEPROM_UPDATED

The system was reset to immediately and automatically force the changes just installed on the MPB-II board and/or one or more SMM4 boards in the DI. A dump is never taken when a reset 46 occurs.

Operational (OP)

The probable cause for each of these resets is something that can most likely be corrected on-site in the software or environmental conditions. The following suggested actions should be taken.

1B(16) = NO_CONFIGURATION_FILE_OBTAINED

Verify proper DI SYSTEM_ID at location 8400(16) by putting the DI in maintenance mode and using the DI console (see the CDCNET Hardware Installation and Troubleshooting manual). If your network is operating under NOS/VE, issue the ACTIVATE_NETWORK_FILE_ACCESS command. If there is no configuration file for the CDCNET system (or if it is busy or otherwise unavailable), an error is reported on the NOS/VE system job log display. Also, inspect the OCU library for a configuration file with the appropriate system identifier.

Under NOS, verify that NETFS is running properly by examining the NAM K-display. Also, use NETFM to list or attach the configuration file (using the NF parameter).

If no configuration file exists, create one. See the CDCNET Configuration Guide.

1C(16) = CONFIGURATION_FILE_READ_ERROR

A configuration file read error occurred or the host file server became unavailable. Check the configuration file and check the status of the file server. If your network is operating under NOS/VE, examine the system job log display to determine if Network File Access restarted or terminated abnormally.

Under NOS, verify that NETFS is running properly by examining the NAM K-display.

2A(16) = STORAGE_ALLOCATION_FAILED

This indicates that not enough memory was available when the Initial Loader was building the loader data structures for a module. Add more memory or remove modules from the boot file before reloading. This reset code is also listed under Load File (LF) action.

38(16) = AC_LOW_ERROR,
39(16) = TEMPERATURE_SHUTDOWN_ERROR

Environmental problems are suspect. Contact installation management personnel or customer engineers.

3D(16) = BOOT_FILE_MEDIA_MISMATCH

The boot file type loaded in the DI did not match the medium it was loaded across; for example, a channel boot file was loaded over ESCI instead of a channel. Look at field boot_map_entry_address in MPB RAM to find out what medium the DI was loaded across. See also this reset under the Load File (LF) heading.

41(16) = MPB_RAM_PTR_NOT_FOUND

The system MPB_RAM_PTR entry point (in EXEC_MPB or ICA_EXEC_MPB) is missing from the boot file. Rebuild the boot file, adding this module, before reloading.

42(16) = TIMER_TASK_MODULE

The TIMER_TASK_MODULE (EXEC_PMM or ICA_EXEC_MPB) is missing from the boot file. Rebuild the boot file, adding this module, before loading.

45(16) = CONFIGURATION_CMD_READ_ERROR

A configuration command read error occurred or the host file server became unavailable while a configuration file command was executing. Check the configuration file and check the status of the file server. Additionally, the log messages in the dump file should show which command this error occurred on. If your network is operating under NOS/VE, examine the system job log display to determine if Network File Access restarted or terminated normally.

Under NOS, verify that NETFS is running properly by examining the NAM K-display.

Load File (LF)

The probable cause for each of these resets is a bad load file. If the load file has never been used successfully before, get a correct file. If this load file has been used successfully before, a software or hardware problem is likely. The following descriptions assume the latter to be true.

10(16) = LOAD_SOFTWARE_TOO_BIG

The boot file is too large to fit into SMM. Remove unnecessary modules from the boot file library or add more SMM before reloading.

This reset may also indicate that the on-board diagnostics detected an SMM failure and have marked a block of SMM as unavailable. The remaining SMM is not sufficient for loading the boot file.

11(16) = IMPROPER_FIRST_MODULE

The first module in the boot file was not the Initial Loader (INITLDRABS). Check the boot file for irregularities, or to see if the library file might have been moved into a boot file by mistake.

12(16) = UNSATISFIED_EXTERNAL

The initial load failed because an entry point was referenced that was not externally declared by any module in the boot file. Missing entry point names are displayed on the DI console. Do a test link (using SES procedure) of the boot file object library after deleting any ABS modules.

13(16) = SYSCONFIG_NOT_LOADED

The SYS_CNFG table (in EXEC_MPB) is missing from the boot file.

14(16) = POST_LOAD_ROUTINES_NOT_FOUND

The entry INITIALIZE_EXECUTIVE (in POST_LOADER_PROCESSING) is missing from the boot file.

16(16) = UNRECOGNIZABLE_OBJECT_TEXT,
1E(16) = IDENTIFICATION_RECORD_EXPECTED,
1F(16) = UNEXPECTED_IDR_ENCOUNTERED,
20(16) = PREMATURE_EOF_ON_FILE,
21(16) = ABSOLUTE_LENGTH_TOO_LARGE,
22(16) = INVALID_OBJECT_TEXT_VERSION,
23(16) = INVALID_MODULE_KIND,
24(16) = INVALID_MODULE_ATTRIBUTE,
25(16) = INVALID_SECTION_ORDINAL,
26(16) = DUPLICATE_SECTION,
27(16) = INVALID_SECTION_KIND,
28(16) = INVALID_ALLOCATION_ALIGNMENT,
29(16) = INVALID_OFFSET,
2B(16) = UNDEFINED_SECTION,
2C(16) = REFERENCE_OUTSIDE_OF_SECTION,
2D(16) = INVALID_ADDRESS_KIND,
2E(16) = INVALID_NUMBER_OF_BYTES_SPANNED,
2F(16) = TRANSFER_SYM_ENTRY_PT_NOT_FOUND

Unknown or unsupported loader text records were found in the boot file or loader library. Check the file module library for irregularities. Check whether newly added modules were compiled with DIDEBUG on or by the wrong compiler or assembler.

17(16) = DUPLICATE_ENTRY_POINT

A duplicate entry point was detected. Do a test link (using SES procedure) of the boot file module library after deleting any ABS modules.

1D(16) = NOT_ENOUGH_MEMORY_FOR_BUFFERS

There must be enough memory after the initial load for allocation of 100 descriptor buffers and 65535 bytes of data buffers. If not, this reset code is issued. Remove unnecessary modules from boot file library before reloading.

2A(16) = STORAGE_ALLOCATION_FAILED

This indicates that not enough memory was available when the Initial Loader was building the loader data structures for a module. Add more memory or remove modules from the boot file before reloading. This reset code is also listed under Operational (OP) action.

30(16) = PARAMETER_VERIFICATION_ERROR

A compilation-time error was detected: the named procedure XDCL and XREF parameters don't match either in type or number. Types must match exactly; they may not match by synonyms or aliases. Assembler entry points must precede CYBIL references when the CYBIL references do not agree.

31(16) = LOADER_TABLE_NOT_FOUND

A crucial loader data structure was not found. This structure is in module OLL_PROGRAM_INTERFACE_PROCS. Under NOS/VE, use the DISPLAY_OBJECT_LIBRARY command to examine the boot file.

Under NOS, do a test link (using the SES.LINK68K procedure) on the object file to determine whether this module is in the boot file.

If the module is missing, add it before reloading.

35(16) = MODULE_CHECKSUM_IS_INVALID

SYSTEM_AUDITS aborted the system because a loaded module was corrupted. Use the Dump Analyzer to examine the SYSTEM_AUDITS stack for the module name and section ordinal, then use DISM to examine the affected memory for recognizable patterns. To locate the SYSTEM_AUDITS stack:

- Use DISTCB, with TI=ALL.
- Examine the output for task name SYSTEM_AUDITS.
- Display stack length number of bytes from the stack segment address in the SYSTEM_AUDITS TCB. This is the SYSTEM_AUDITS stack.

3C(16) = SYSTEM_DATA_NOT_FOUND

The SYSTEM_DATA entry point (in SYSTEM_AUDITS) is missing from the boot file. Rebuild the boot file, adding this module before reloading.

3D(16) = BOOT_FILE_MEDIA_MISMATCH

The boot file type loaded in the DI did not match the medium it was loaded across; for example, a channel boot file was loaded over ESCI instead of a channel. Look at field boot_map_entry_address in MPB RAM to find out what medium the DI was loaded across. Look at the XDCL'ed variable abort_message to see what the boot file type is.

47(16) = LOADER_BUS_ERROR

A bus error occurred during the initial load sequence. If any SMM errors occurred while leaving on-board diagnostics, the fault LED remains lit on the respective board(s).

Hardware (HW)

A hardware problem is the probable cause for each of these resets. Perform the suggested hardware problem isolation or correction.

03(16) = HALT_MEMORY_FAULT

A double-bit SMM error occurred. Check NPA reports to identify failing SMM board.

Board failure might show on indicator light if onboard diagnostics failed; this is seen after the DI resets and is going through diagnostics. See the CDCNET Hardware Installation and Troubleshooting manual.

08(16) = SRAM_PARITY_ERROR_RESET

A parity error occurred reading MPB-II causing a hardware reset. The reset recovery register indicates to the onboard diagnostics that a SRAM parity error was the cause of the reset. The contents of the reset recovery register is saved in the card map table. The reset recovery register also contains the memory bank which has the parity error on the byte. The failure management software logs information pertaining to the SRAM parity error at system restart.

37(16) = SMM_DOUBLE_BIT_ERROR

A double-bit SMM error occurred. Try the reset or power-on diagnostics to isolate the failing SMM board. See the CDCNET Hardware Installation and Troubleshooting manual.

3F(16) = HARD_FAILURE

This reset code is part of CDCNET's failure management feature. A hard failure is defined as one from which recovery is not possible; it can be caused by hardware or software.

In the error log file, this message indicates the board slot number for the DI subsystem where the failure occurred. A separate log message is issued for the failing subsystem or its failed software. Examine the executive error table for clues.

Software (SW)

A software bug is the probable cause for each of these resets. Submit a PSR with the dump file and CDCNET log file.

04(16) = DEAD_MAN_TIME_OUT

A running task took too long to execute, preventing SYSTEM_AUDITS from resetting the timer.

Use the Dump Analyzer to determine why the task timed out:

1. Use the DISTCB subcommand to identify the task with the task state RUNNING—this is the task that timed out.

2. Use the DISSCT subcommand to look at the following values:

   INTERRUPT FIREWALL CHAIN ADDRESS. This identifies the interrupt processor.

   BINARY TIME-OF-DAY. This indicates the millisecond clock value at the time of failure.

3. Use the DISM subcommand to look at the LAST_DEADMAN_RESET value in the system data record.

If the difference between the binary time-of-day and LAST_DEADMAN_RESET is less than 10,000(10) and the interrupt firewall chain address was 0, then the error is probably hardware-related. Try to isolate the problem using the CDCNET Hardware Installation and Troubleshooting manual before writing a PSR.

18(16) = TASK_ERROR_NO_RECOVERY_PROC,
19(16) = TASK_ERROR_EXCEED_MAX_RECOVERS,
1A(16) = TASK_ERROR_UNRECOVERABLE,
36(16) = SOFTWARE_DEAD_STOP

The task that caused the reset has a task state of RUNNING or SUSPEND.

34(16) = STOP_EXECUTIVE

Use the Dump Analyzer DISEET or DISM subcommands to examine the executive error table for error information. The field STOP_SUPERVISOR_STACK_POINTER contains the supervisor stack pointer at the time of the reset. Using DISM, display this stack. The top of the stack contains the return address to the caller of STOP_EXEC.

3B(16) = OVERFLOWED_STACK

A stack overflow was detected on a call to the Executive when the value of register A7 was numerically less than the first byte address of the stack, or by SYSTEM_AUDITS after a task had stopped. Use the VALSA subcommand to reveal violations of stack areas. Or, use DISC to check the affected task for recursive calling. Move large variables off the stack or increase the stack size. Use ALLOCATE/FREE rather than PUSH CYBIL statements, if feasible.

3E(16) = CYBIL_DETECTED_ERROR

CYBIL run-time routines detected an error (when compiled with range checking on). Correct code and rebuild the boot file before reloading.

3F(16) = HARD_FAILURE

This reset code is part of CDCNET's failure management feature. A hard failure is defined as one from which recovery is not possible; it can be caused by hardware or software. See the description of this reset code under the Hardware (HW) heading.

43(16) = TASK_RECEIVED_UNKNOWN_ITM

This reset code indicates that an unknown ITM task message was received by a task.

44(16) = SNA_3270_TIP_DHCF_ABORT

This reset code occurs when the TIP software traps an invalid request response unit.

Procedures to Enhance Operator Environment D

This appendix describes a number of NOS/VE and NOS batch jobs and procedures that facilitate CDCNET network management. The batch jobs are part of the CDCNET product. The procedures are not part of the CDCNET product and PSRs cannot be written against them. CYBER Software Support will answer questions you may have about them. These procedures have been used internally to manage CDCNET and we are now are making them available for our customers to use as well.

CDCNET Network Management Procedures for NOS/VE

The following is a summary of the jobs and procedures available for use on NOS/VE. The batch jobs are part of the CDCNET product. The SCL procedures are in the answer text for PSR CSFA094. The *EOR separates procedures from one another. Log on to SOLVER to get the source for the procedures.

For more information on SCL procedures, see the NOS/VE System Usage Manual.

The jobs and procedures available under NOS/VE to help CDCNET network management include:

- BROADCAST_CONFIGURATION_FILES
- CREATE_COMMAND_CONNECTION
- DISPLAY_PHYSICAL_NAMES
- DISPLAY_SYSTEM_NAMES
- INFORM_USERS
- PROCESS_DUMP_FILES
- SEND_COMMAND_EVERYWHERE
- PROCESS_LOG_JOB

BROADCAST_CONFIGURATION_FILES Procedure

This procedure broadcasts a file containing CDCNET configuration procedures to one or more NOS/VE systems and uses the standard CDCNET procedure REPLACE_CONFIGURATION_FILE to replace the procedures on each system. This allows your site to maintain equivalent configuration files on all machines in the CDCNET network. The QTF/PTF product is required to run this procedure.

CREATE_COMMAND_CONNECTION Procedure

This procedure allows you to select a group of DIs to work with. Once selected, all following commands are sent to those DIs without entering the SEND_COMMAND command. Only the command itself needs to be entered.

DISPLAY_PHYSICAL_NAMES Procedure

This procedure displays CDCNET device names such as $DI_ and $ICA_ prefixed names.

DISPLAY_SYSTEM_NAMES Procedure

This procedure displays the names in the network typically registered by the CDCNET DEFINE_SYSTEM command. The content of the display is determined by the title pattern specified by the SYSTEM parameter.

INFORM_USERS Procedure

This procedure accepts an infinite number of message lines and then uses the WRITE_TERMINAL_MESSAGE command to send the message to CDCNET users with active connections. This message can be restricted to users of a particular service or those connected to DIs with a particular title pattern. This procedure resides on $SYSTEM.OSF$SITE_COMMAND_LIBRARY.

PROCESS_DUMP_FILES Procedure

This procedure allows you to select a group of dump files and to perform management functions on those files. The following menu displays for each dump file:

Press	To
A	Analyze dump
B	Save dump to a backup file
C	Copy dump
D	Delete dump and select another
M	Move dump to an administrator's catalog
V	Select dump analyzer version
W	Write summary dump analysis to file
<cr>	Select another dump
QUIT	Stop processing dumps
anything else	Process as a NOS/VE command

The procedure allows management of the dump files without being familiar with the dump analyzer or dump file catalog structure.

SEND_COMMAND_EVERYWHERE Procedure

This procedure allows you to send any CDCNET command to all DIs/ICAs or to a group of DIs/ICAs with a particular title pattern.

PROCESS_LOG_JOB (NPA)

The NPA process log job consists of the following three separate jobs:

PROCESS_LOG_JOB
GENERATE_NPA_REPORTS
ARCHIVE_NPA_DATABASES

All three jobs reside in file $SYSTEM.CDCNET.VERSION_INDEPENDENT.PROCESS_LOG_JOB. They are delimitted by nested JOB/JOBEND statements. NOS/VE Network Log Management submits the PROCESS_LOG_JOB, which, depending on conditions, may create and submit the GENERATE_NPA_REPORTS job. This in turn creates and submits the ARCHIVE_NPA_DATABASES job.

PROCESS_LOG_JOB

This batch job resides in file $SYSTEM.CDCNET.VERSION_INDEPENDENT.PROCESS_LOG_JOB. It is submitted by NOS/VE Network Log Management and performs the following functions:

1. Reformats all inactive cycles of the CDCNET log file to NPA databases in catalog $SYSTEM.CDCNET.ANALYSIS.

2. Copies all processed cycles of $SYSTEM.CDCNET.LOG to file $SYSTEM.CDCNET.PROCESSED_LOG_FILES and deletes the originals.

3. Submits the GENERATE_NPA_REPORTS job once per day to generate NPA reports for the previous day and back up $SYSTEM.CDCNET.PROCESSED_LOG_FILES to tape.

This job saves its output and job log to file $SYSTEM.CDCNET.ANALYSIS.PROCESS_LOG_JOB_OUTPUT. No output is printed unless a problem occurs.

GENERATE_NPA_REPORTS

This batch job originates as text within the PROCESS_LOG_JOB batch job. It is submitted by the PROCESS_LOG_JOB batch job and performs the following functions:

1. Generates selected NPA reports, saving them in catalog $SYSTEM.CDCNET.ANALYSIS.CURRENT_REPORTS. Existing copies of reports are deleted.

2. Saves the date on which reports were last generated on file $SYSTEM.CDCNET.ANALYSIS.LAST_REPORT_DATE. This file is used by the PROCESS_LOG_JOB batch job to decide when to submit the report generator job.

3. Backs up processed raw log file(s) ($SYSTEM.CDCNET.PROCESSED_LOG_FILES) to tape and deletes the file(s).

4. Saves the date of last backup in file $SYSTEM.CDCNET.LAST_BACKUP_DATE. This is used to prevent accidental backup twice in one day (which would result in loss of data) should the job be run more than once.

5. Submits the ARCHIVE_NPA_DATABASES batch job to clean up databases.

This job saves its output and job log to file $SYSTEM.CDCNET.ANALYSIS.GENERATE_NPA_REPORTS_OUTPUT.

ARCHIVE_NPA_DATABASES

This batch job originates as text within the GENERATE_NPA_REPORTS batch job. It is submitted by the GENERATE_NPA_REPORTS batch job and performs the following functions:

1. Archives the data in the NPA databases residing in catalog $SYSTEM.CDCNET.ANALYSIS.

2. Eliminates old data from the NPA databases and discards it.

This job saves its output and job log to file $SYSTEM.CDCNET.ANALYSIS.ARCHIVE_NPA_DATABASES_OUTPUT. No output is printed unless a problem occurs.

CDCNET Network Management Procedures for NOS

The following is a summary of the procedures and jobs available under NOS. These procedures assume a file PROCFIL resides on user name SYSUN on the default family. This user name must be validated for secondary user commands. Many of these procedures require XEDIT.

Log on to SOLVER to get the source for these procedures and batch jobs. They are in the answer text for PSR CSFA081. The *EOR separates procedures from one another.

The procedures and jobs available under NOS to help CDCNET network management include:

- NPAARC
- NPANLA
- NPANLR
- NPANLT
- NPAREPS

NPAARC Procedure

This procedure is in file NPAARC on username NETADMN along with a FORTRAN program that calculates which databases are more than four days old. Database information older than four days is archived (data deleted).

NPANLA (Batch Job)

This batch job is in file NPANLA on username NETADMN. The main function of NPANLA is to call the procedure NPAARC which archives databases. This job is submitted daily by NPAREPS.

NPANLR (Batch Job)

This batch job is in file NPANLR on username SYSUN. NPANLR is submitted from the system startup procedure. It reformats the previously terminated log file; executes NPANLP to purge log files; and submits another job, NPAREPS; which runs NPA reports.

NPANLT Procedure

This procedure is in file PROCFIL on username SYSUN. NPANLT should be called from the system closedown procedure once per day. NPANLT terminates the active log file and copies all processed raw log files to tape. It also creates a procedure (NPANLP/UN=SYSTEMX) that deletes log files successfully dumped to tape (to be executed later).

NPAREPS (Batch Job)

This batch job is in file NPAREPS on username NETADMN. NPAREPS generates NPA reports and writes them to a file NETREPT on username NETADMN.

It then submits another batch job, NPANLA, to archive databases.

MPB Memory Map E

This appendix maps the DI's Main Processor Board (MPB) memory. The addresses given are not offsets, but actual DI memory locations. Figure E-1 shows an overall picture of MPB board slot addressing.

Address	Slot
03FFFF_	
040000	#0
050000	#1
060000	#2
070000	#3
080000	#4
090000	#5
0A0000	#6
0B0000	#7
0C0000	#8
0D0000	#9
0E0000	#10
0F0000	#11

CARD_SLOTS

The MPB may access 16x64K byte segments of non-SMM space. Four of these segments are assigned permanently as MPB local address space. The remaining 12 slots are provided for direct peripheral memory access by the MPB. The board slots logically occupy 050000 through 0FFFFF. The numbers allocated represent the physical location or board number to be assigned during installation. Each slot contains 64K bytes of contiguous address space which, when addressed, allow the associated board to logically connect its address bus to the Internal System Bus (ISB).

Only MCI boards allow direct access. Whether the board allows access or not is determined during startup via the ISB control bus. Attempted access to a nonconfigured board evokes the same response as the attempt to address any nonexisting address; that is, an instruction timeout error is issued to the component attempting the access.

The local RAM of an intelligent peripheral is placed in a dump file at the corresponding board slot address. For example, if a CIM board was installed in slot #5, its memory can be displayed with the following Dump Analyzer subcommand:

```
DISM A=90000(16) BC=6000(16)
```

Figure E-1. Board Slot Addressing

MPB RAM Tables

Table E-1 provides information about fixed address memory within MPB RAM. Memory at these addresses is battery-backed. If the DI's AC power source is interrupted for any reason, memory at these addresses is protected and is used to reinitialize the DI.

The offsets given below are actual DI addresses, expressed in hexadecimal.

Table E-1. MPB RAM Tables

Offset	Field Name	Description
+0	vector	An array of pointers to cells that constitute the vector space
+400	system_id	Unique identifier for this hardware box
+406	system_id_checksum	System_id checksum
+408	table_format_version	Version of RAM table format
+40A	status	The MPB status register is found in lower four-bits
+40B	mpb_ram_zeroed	A flag that indicates if MPB RAM has been zeroed (box was powered-on)
+40C	smm_size	The number of contiguous usable SMM bytes from address 100000(16)
+410	boot_map_entry_address	A pointer to the map entry used as the bootstrap board
+414	auto_dump_table_address	A pointer to the Auto Dump Table
+418	reset_status	The reset status saved from the most recent reset
+419	reset_code	The reset code (from both software and hardware)
+41A	software_error_code	The software error code
+41B	hardware_reset_code	The possible hardware cause for reset
+41C	version	Version code found within last accepted help offer PDU
+41E	network_id	The network identifier found within the last accepted help offer PDU
+422	help_system_id	The system identifier found within the last accepted help offer PDU
+428	auto_dump_subroutine_address	A pointer to the routine that generates the Auto Dump Table

(Continued)

Table E-1. MPB RAM Tables *(Continued)*

Offset	Field Name	Description
+42C	auto_dump_subroutine_length	Length of the routine that generates the Auto Dump Table, in 16-bit words
+42E	auto_dump_subroutine_checksum	A checksum of the routine that generates the Auto Dump Table
+430	map_table	The Board Map Table is described later in this appendix
+550	reserved_4_bytes	Reserved for future use
+554	mpb_error_routine_pointer	A pointer to the MPB error routine
+558	mpb_error_routine_length	The length of the MPB error routine, in 16-bit words
+55A	pmm_error_routine_pointer	A pointer to the PMM error routine
+55E	pmm_error_routine_length	The length of the PMM error routine, in 16-bit words
+560	smm_error_routine_pointer	A pointer to the SMM error routine
+564	smm_error_routine_length	The length of the SMM error routine, in 16-bit words
+566	expected_smm_interrupt_flag	A pointer to the expected SMM interrupt flag
+56A	ept_address	A pointer to the entry point table
+56E	loaded_module_list	A pointer to first loaded module entry
+57A	unsatisfied_externals	A pointer to the unsatisfied externals table
+57E	checksum_value	Checksum value
+580	checksum_length	Length of checksummed area in words
+582	checksum_format	Currently unused
+584	desbuflen	The configured length of descriptor buffers
+588	datbuflen	The configured length of data buffers
+58C	reserved_memory	Reserved memory, critical use
+58E	master_date	Master date for system
+592	reset_time	Time at last system reset
+596	date_initialized	Flag indicating if master_date or default date value should be used

(Continued)

Table E-1. MPB RAM Tables *(Continued)*

Offset	Field Name	Description
+597	time_initialized	Flag indicating if the real-time clock or default time value should be used
+598	max_pmm_required	Maximum PMM required
+59C	channel_max_frame_sizes	Number of slots in a DI
+5AC	cdcnet_nsap_address_prefix	CDCNET address prefix type
+5B7	cdcnet_nsap_address_size	CDCNET address size range
+5B7:5	pad_for_nsap_addr_size	Unused
+5B8	cdcnet_nsap_address_prefix_size	Size of the CDCNET address prefix
+5B8:4	pad_for_nsap_addr_prefix_size	Unused
+5B9	osi_protocol_stack_enabled	True if OSI protocol enabled
+5BA	unused_carry_over_from_r142	Unused 1 byte
+5BB	preferred_protocol_stack	Preferred protocol is either OSI or XNS
+5BB:1	mpb_ram_initialized	If true, MPB RAM is initialized
+5BB:2	pad_for_preferred_stack	Unused
+5BC	sys_cnfg_ptr	A pointer to Executive's System Configuration Table
+5C0	system_ancestor_task_id	A pointer to the system ancestor TCB
+5C4	current_transport_selector	Next transport_selector
+5C6	data_unit_identifier	Data unit identifier for CLNS
+5C8	unused_carry_over_from_r142	Unused 2 bytes
+5CA	initial_loader_active	True if initial loader is active
+5CC	mpb2_reset_recovery_reg	Reset recovery register
+5CE	lwb_filler	Two byte filler
+5DO	system_failure_table	The system failure table
+682	reset_40_array	Reset reason for 4 latest resets
+684	reset_40_array_index	Index to array (a word must be filled up)
+686	mpb_end	First header of MPB extent chain

Board Map Table

From byte offset 430(16) to 550(16) in fixed-address memory, there is a table used to record information about the modular hardware installed in a DI. Space is reserved in this table for each of the eight major board slots, as follows:

Board Slot Number	0	1	2	3	4	5	6	7
Starting Address	430	454	478	49C	4C0	4E4	508	52C

The first 14 bytes of each entry in this table have the same structure. The structure of bytes 15 through 24 of each entry depends on the type of board installed in that slot.

Board Map Common Information

Table E-2 gives descriptions of the fields found in the first 14 bytes of any board map entry.

Table E-2. Board Map Common Information

Offset	Field Name	Description
+0	slot_number	The main board slot number of this board
+1	icb_read_reg_zero	The contents of ICB read register zero (see table E-3)
+2	icb_write_register_zero	The contents of ICB write register zero
+3	icb_write_register_one	The contents of ICB write register one
+4	qck_lk_diag_status	Quicklook diagnostics status; if not zero, testing failed
+6	card_version	Name code extension for this board
+7	status_extension	Status extension for this board
+8	intllgnt_card	Intelligent board flag; set to zero if the board is not an intelligent peripheral
+9	byte_wide	Set to zero if board is not byte wide ROM
+A	id_rom_tbl_address	A pointer to the ROM table; set to zero if none exists
+E	icb_address	The ICB address for this slot
+10	itb_address	The ITB address for this slot
+14	See Board Map Type-Specific Information next in this appendix	Board type-specific fields; see to Board Map Type-Specific Information in this appendix

Table E-3. ICB Read Register Zero

Bit(s)	Description
7	If 0, local secondary power bad or quicklook phase 1 failed; if 1, no fault on board
6	If 0, device not available; if 1, device available
5	If 0, attention switch is set to off; if 1, attention switch on
4	If 0, bootstrap not allowed over this board; if 1, bootstrap allowed
3-0	Board code, as follows:

Category	Code	Board
Intelligent Peripheral	0	MPB
	1	CIM
	2	ESCI
	3-7	Reserved
Memory	8	PMM
	9	SMM
	A-B	Reserved
Dump Peripheral	C	DISC
	D	MCI
	E	Reserved
Power on MC in Progress	F	All except MPB

Board Map Type-Specific Information

The last 16 bytes of each board map entry are used to store board type-specific information, regardless of the board type installed.

The board type can be determined from the major card status table (MCST) described in appendix F.

MPB/MPB-II Board

Offset	Field Name	Description
+14	mpb_error_count	The number of MPB errors since last reset
+16	MPB-II_SCSI_error	Non-zero value indicates an SCSI failure. These are treated as non-fatal errors.
+17	mpb_short_test	Non-zero indicates that full power-up diagnostics were not run due to MPB switch 5 being ON
+18	MPB-II_ram_parity_error	Contents of the Reset Recovery Register for the first recovered RAM MPB-II parity error Bit 9 - 1 = RAM Parity Error Reset Bit 4 - 0 = Bank 0 1 = Bank 1 Bit 3 - 1 = Byte 3 Parity Error Bit 2 - 1 = Byte 2 Parity Error Bit 1 - 1 = Byte 1 Parity Error Bit 0 - 1 = Byte 0 Parity Error
+1A	mpb_ram_address	A pointer to the first failing MPB RAM location. The address given is biased by 8000(16) because RAM and ROM are swapped during error testing.
+1E	mpb_ram_errors	The number of MPB RAM read parity errors
+20	unused	Bytes 21 through 24 are not used in an MPB entry

ESCI Board

Offset	Field Name	Description
+14	esci_reserved_status	ESCI board status; reserved area
+15	esci_transceiver_status	Transceiver status flag

PMM Board

Offset	Field Name	Description
+14	pmm_mem_blk_size	The size of each PMM memory block
+15	filler_2	Not used
+1A	pmm_address	A pointer to the first failing PMM address
+1E	pmm_errors	The number of PMM read parity errors
+20	unused	Bytes 21 through 24 are not used in a PMM entry

SMM Board

Offset	Field Name	Description
+14	smm_mem_blk_size	The size of each SMM memory block. Shift left 16 bits for size in bytes; 8 = 512 K; 20 = 2 M
+15	smm_start_blk_0	The starting address of SMM block zero; if zero, unavailable
+16	smm_start_blk_1	The starting address of SMM block 1
+17	smm_start_block_2	The starting address of SMM block 2
+18	smm_start_block_3	The starting address of SMM block 3
+19	filler_3	Not used
+1A	smm_sbe_log	The contents of the error log for the first SMM single-bit error
+1C	smm_sbe_errors	The number of SMM single-bit errors
+1E	smm_mbe_log	The contents of the error log for first SMM multi-bit error
+20	smm_mbe_errors	The number of SMM multi-bit errors
+22	smm_rom_version	The SMM board ROM version number

CIM Board

Offset	Field Name	Description
+14	lim	LIM configuration table; 2 bytes per LIM, indicating type and status

MPB-II Memory Map and Address Mapping

On the MPB-II there is a Paged Memory Management Unit (PMMU). The PMMU provides address translation and memory protection for a demand-paged virtual memory system. In figure E-2, the logical addresses, when accessed, are mapped to the given physical addresses by the PMMU.

The MPB-II memory map is equivalent to the MPB memory map with the following exceptions (comparisons are made between MPB-II and MPB.).

- Local MPB RAM on the MPB-II is located at $1000000 to $0107FFFF. Addresses above $FFFFFF do not exist on the MPB (i.e. the MC68000 has only a 24-bit address bus). On the MPB, local RAM is located from $00000000 to $00003FFF.

- Note that on the MPB-II logical addresses $00000000 to $00003FFF and $01000000 to $010003FFF are both mapped to physical addresses $01000000 to $01003FFF to provide compatibility between MPB post ROM/RAM swap and MPB-II.

- On the MPB-II, logical addresses $00008000 to $0000FFFF are used for the Electronically Erasable Programmable Read Only Memory (EEPROM). On the MPB, addresses $00008000 to $0000BFFF are used for MPB ROM; however addresses $0000C000 to $0000DFFF are unused and $0000E000 to $0000FFFF is local I/O. Note that MPB Memory Mapped Input/Output is equivalent with the MPB-II local I/O addresses.

- The Internal Transfer Bus (ITB) and Card Slot addressing is equivalent on the MPB and MPB-II. However, on the MPB-II the local I/O is expanded to include addressing for the SCSI device.

- On the MPB-II, logical addresses $01080000 to $01083FFF are mapped to physical addresses $01004000 to $01007FFF. Logical addresses $01004000 to $01007FFF are not mapped so that Executive stack underflow results in a bus error.

- On the MPB-II, logical addresses $0108C000 to $0108FFFF are used for the Logic Cell Array Electronically Erasable Programmable Read Only Memory (LCA EEPROM). The LCA EEPROM contains the configuration data for the XILINX logic cell arrays which contain the bulk of the random logic on the MPB-II.

- On the MPB-II, the logical address sections from $00000000 to $00003FFF and from $0108C000 to $0108FFFF are only accessible in supervisor state. Accesses to these ranges in user state result in bus errors.

- The following logical address ranges are write-protected from user state programs: $00000000 to $00003FFF, $00008000 to $0000BFFF, $0000C000 to $0000FFFF, $01000000 to $01003FFF and $0108C000 to $0108FFFF.

- The Program EEPROM and LCA EEPROM on the MPB-II can be automatically updated by software.

Logical Address		Physical Address
\$00000000 \$00003FFF	Local RAM	\$01000000 \$01003FFF
\$00004000 \$00005FFF	Not Used	\$00004000 \$00005FFF
\$00006000 \$00007FFF	Local I/O	\$00006000 \$00007FFF
\$00008000 \$0000BFFF	Program EEPROM #1	\$00000000 \$00003FFF
\$0000C000 \$0000FFFF	Program EEPROM #2	\$00008000 \$0000BFFF
\$00010000 \$0003FFFF	Not Used	Undefined Bus Error
\$00040000 \$000FFFFF	Card Slot Address	\$00040000 \$000FFFFF
\$00100000 \$00FFFFFF	Internal Transfer Bus	\$00100000 \$00FFFFFF
\$01000000 \$01003FFF	Local RAM	\$01000000 \$01003FFF
\$01004000 \$01007FFF	Local RAM	Exec Stack Underflow Bus Error
\$01008000 \$0107FFFF	Local RAM	\$01008000 \$0107FFFF
\$01080000 \$01083FFF	Local RAM	\$01004000 \$01007FFF
\$01084000 \$0108BFFF	Not Used	Undefined Bus Error
\$0108C000 \$0108FFFF	Logic Cell Array EEPROM	\$0000C000 \$0000FFFF
\$01090000 \$1FFFFFFF	Not Used	Undefined Bus Error
\$20000000 \$FFFFFFFF	Not Used	Variant results because the upper seven bits are ignored

Figure E-2. MBP-II Memory Map

System Tables F

This appendix describes several tables and control blocks maintained in DI memory for use by software components. They are:

- Executive error table including the error buffer
- System configuration table
- System data record
- Hardware status tables
  - Major card status table
  - LIM status table
  - Port status table
  - SMM bank status table
  - PMM bank status table
  - Link information block
- Timer queue
- Directory data stores
  - Registration data store
  - Translation data store
  - Translation request data store
- Routing data stores
  - Least cost routing data store
  - Local DCN data store
  - Received DCN data store
- Terminal support debug table
- Batch data service debug table
- Batch gateway debug table
- Operator support application table
- Loader entry point table
- System memory management table
- Tree root structure

NOTE

All data structures described in this appendix are valid for CDCNET Version 1.4 and above. All offsets are quoted in hexadecimal.

Executive Error Table

The executive error table is initialized by the DI Executive and is used to maintain information regarding software errors. Preselected fields from this table can be displayed using the Dump Analyzer DISEET subcommand.

The first 28 bytes of the executive error table record general DI error information. These fields are followed by a number of error buffers, one for each error recorded in the executive error table. The total length of the executive error table structure is 28(16) + (8C(16) * # error buffers).

An executive error table that contains one error buffer can be displayed from a dump file using the following Dump Analyzer subcommand:

```
DISM A=EXEC_ERROR_TABLE RC=0B4
```

Table F-1 summarizes the fields in the executive error table. Table F-2 describes the error buffers. All offsets are expressed in hexadecimal. An example showing an exploded view of each table follows table F-4.

Table F-1. Executive Error Table

Offset	Field Name	Description
+0	stop_supervisor_stack_pointer	Executive stop supervisor stack pointer
+4	last_error_address	Pointer to last error
+8	lock_last_error	Last error pointer being updated
+0A	address_error_being_processed	Flag: address error being processed
+0C	number_of_spurious_interrupts	Number of spurious interrupts
+0E	smm_error_count	Number of single-bit SMM errors; two bytes per card
+1E	total_error_count	Total errors inserted into buffers
+20	system_ancestor_tcb	Pointer to System Ancestor TCB for error intertask messages
+24	debug_address_called_on_error	Address of DI Debugger, called on error (assumes DI Debugger loaded)
+28	error_buffers	Error information structures; see table F-2

NOTE

Error buffers are not necessarily in chronological order.

Table F-2. Error Buffers

Offset	Field Name	Description
+0	executive_error_code	Type of error; see executive_error_ codes below
+2	lock_error_buffer	Buffer locked for processing
+4	binclock_at_time_of_error	Binary clock value at time of error
+8	d0_thru_d7	Registers D0 through D7; 4 bytes per register
+28	a0_thru_a6	Registers A0 through A6; 4 bytes per register
+44	status_register	Status register
+46	supervisor_stack_pointer	Supervisor stack pointer
+4A	user_stack_pointer	User stack pointer
+4E	program_counter	Pointer to program counter 2 through 10 bytes past instruction causing the error
+52	tcb_for_running_task	Pointer to TCB of task running at time of error
+56	module_name	Name of module of task running at time of error
+76	module_offset	Offset in module of task running at time of error
+78	error_during_firewall	Flag: error during firewall
+7A	firewall_procedure_address	Firewall procedure address
+7E	mpb_status_register	MPB status register, case exec error codes of bus and address errors
+80	CASE exec_error_codes	See tables F-3 and F-4

Executive_Error_Codes

00(16) = unused_0	0A(16) = line_1010_interrupt_i
01(16) = unused_1	0B(16) = line_1111_interrupt_i
02(16) = bus_error_i	0C(16) = smm_single_bit_error_i
03(16) = address_error_i	0D(16) = smm_double_bit_error_i
04(16) = illegal_instruction_i	0E(16) = task_runs_too_long_i
05(16) = zero_divide_i	0F(16) = smm_bus_parity_error_i
06(16) = chk_instruction_i	10(16) = format_error
07(16) = trapv_instruction_i	11(16) = mmu_config_error_i
08(16) = privilege_violation_i	12(16) = dead_man_timeout_i
09(16) = trace_interrupt_i	13(16) =manual_reset_i

Table F-3. CASE exec_error_codes of bus_error_i, address_error_i for MPB

Offset	Field Name	Description
+80	first_failure_capture_address	First failure capture address
+84	bus_exception_status	Bus/address exception status
+86	access_address	Access address
+8A	instruction_register	Instruction register

Table F-4. CASE exec_error_codes of smm_single_bit_error_i, smm_double_bit_error_i

Offset	Field Name	Description
+80	smm_card_slot	Major board slot of SMM board
+82	smm_error_log	SMM error log

Table F-5. CASE exec_error_codes of ICA-II

Offset	Field Name	Description
+80	ica_fault_address	Fault address
+84	ica_special_status_word	Special status word
+86	ica_access_address	Access address
+8A	ica_instruction_register	Instruction register

Table F-6. CASE exec_error_codes of bus_error_i, address_error_i for MPB-II

Offset	Field Name	Description
+80	mpb2_nmi_recovery_register	NMI recovery register
+82	mpb2_bus_error_recovery_register	Bus error recovery register
+84	mpb2_cpu_root_ptr_upper	68030 MMU CPU root pointer
+88	mpb2_cpu_root_ptr_lower	68030 MMU CPU root pointer
+8C	mpb2_translation_control	68030 Translation control reg.
+90	mpb2_special_status_word	Special status word
+92	mpb2_instruction_pipe_c	Instruction pipe stage C
+94	mpb2_instruction_pipe_b	Instruction pipe stage B
+96	mpb2_data_fault_address	Data cycle fault address
+9A	mpb2_stage_b_address	Stage B address; long format only

Example:

Following is a section of DI memory that includes the executive error table. This display was made from a dump file using the Dump Analyzer's DISM subcommand. The memory is first presented exactly as it would be displayed by the Dump Analyzer. It is then exploded and tagged for identification of the executive error table fields.

From DISM:

```
STARTING AOORESS:     0A58

HEX ADOR             HEXADECIMAL OATA                      ASCII DATA

     0A58  0000 3FEC 0000 0A80 0000 0000 0000 0000     ?1
     0A68  0000 0000 0000 0000 0000 0000 0000 0001
     0A78  0010 30DC 0000 0000 0003 FFFF 0196 BE2C     =\           >,
     0A88  0000 0001 0000 0184 0000 0004 0000 FFFF
     0A98  0000 0004 0000 0001 0000 FFFF 0010 4996                  I
     0AA8  001A 07CA 001E 0D0F 001A 0C06 001A 0C30      J             0
     0AB8  0000 0964 001A 374A 0002 F7EA 0000 0000      d  7J  wj
     0AC8  3FFE 0002 F78E 0019 DA32 001A 374A 4F50    ?~  w    Z2   7JOP
     0AD8  4552 4154 4F52 5F53 5550 504F 5254 5F41    ERATOR_SUPPORT_A
     0AE8  5050 4C49 4341 5449 4F4E 2020 2020 3486    PPLICATION     4
     0AF8  0000 0000 0000 6980 0000 6107 3361 001E          i    a 3a
     0B08  0D25 337C                                   %3|
```

Executive Error Table, Exploded View:

Hex Address	Hexadecimal Data	Field Name
0A58	0000 3FEC	stop_supervisor_stack_pointer
0A5C	0000 0A80	last_error_address
0A60	0000	lock_last_error
0A62	0000	address_error_being_processed
0A64	0000	number_of_spurious_interrupts
0A66	0000	smm_error_count
0A68	0000 0000 0000 0000 0000 0000 0000	smm_error_count (cont'd)
0A76	0001	total_error_count
0A78	0010 3DDC	system_ancestor_tcb
0A7C	0000 0000	debug_address_called_on_error

Error Buffers, Exploded View:

Hex Address	Hexadecimal Data	Field Name
0A80	0003	executive_error_code
0A82	FFFF	lock_error_buffer
0A84	0196 BE2C	binclock_at_time_of_error
0A88	0000 0001 0000 0184 0000 0004 0000 FFFF	d0_thru_d3
0A98	0000 0004 0000 0001 0000 FFFF 0010 4996	d4_thru_d7
0AA8	001A 07CA 001E 0D0F 001A 0C06 001A 0C30	a0_thru_a3
0AB8	0000 0964 001A 374A 0002 F7EA	a4_thru_a6
0AC4	0000	status_register
0AC6	0000	supervisor_stack_pointer
0AC8	3FFE	supervisor_stack_pointer (cont'd)
0ACA	0002 F78E	user_stack_pointer
0ACE	0019 DA32	program_counter
0AD2	001A 374A	tcb_for_running_task
0AD6	4F50 4552 4154 4F52 5F53 5550 504F 5254	module_from_program_counter
0AE6	5F41 5050 4C49 4341 5449 4F4E 2020 2020	module_from_program_counter (cont'd)
0AF6	3486	offset_into_module
0AF8	0000	error_during_firewall
0AFA	0000 0000	firewall_procedure_address
0AFE	6980	mpb_status_register
0B00	0000 6107	first_failure_capture_address1
0B04	3361	bus_exception_status1
0B06	001E 0D25	access_address1
0B0A	337C	instruction_register

System Configuration Table

The system configuration table is a data structure that retains the status of essential CDCNET system variables. It is a record with fields indicating such things as the highest address in MPB RAM and the states of memory and buffers. Selected fields of the system configuration table in a dump file can be displayed using the DISSCT subcommand.

The entire system configuration table can be displayed from a dump file using the following subcommand:

```
DISM A=SYS_CNFG RC=19E
```

Table F-7 summarizes the fields in the system configuration table. All offsets are expressed in hexadecimal.

Table F-7. System Configuration Table

Offset	Field Name	Description
+0	maxprior	Highest valid priority; lowest is zero
+2	databac	Number of available data buffers
+4	descbac	Number of available descriptor buffers
+6	lwfill01	Fill to keep on long word boundaries
+8	lbufflen	Length of data space, in bytes
+0C	sbufflen	Length of descriptor buffer, in bytes
+10	stdstack	Standard stack allocation
+14	running	Task pointer to running task
+18	curprior	Priority of currently running task
+1A	schprior	Highest scheduled priority
+1C	pmtok	Task preemption permission flag
+1E	lwfill02	Fill to keep on long word boundaries
+20	vecslice	Vector for time slice interrupt
+24	vecintvl	Vector for interval timer interrupt
+28	vecclock	Vector for millisecond interrupt
+2C	mpbramtop	Largest numerical address in MPB RAM
+30	privatetop	Largest numerical address in the private memory module (PMM)
+34	globfree	Number of bytes of free global memory

(Continued)

Table F-7. System Configuration Table *(Continued)*

Offset	Field Name	Description
+38	locfree	Number of bytes of free private memory
+3C	mpbfree	Number of bytes of free MPB RAM memory
+40	globfrag	Number of extents of free global memory
+42	locfrag	Number of extents of free private memory
+44	mpbfrag	Number of extents of free MPB RAM memory
+46	deloadtyp	Type of memory to release flag for deload task
+48	deloadtcb	Task pointer of deload task
+4C	deloadmpb	Deloadable bytes of MPB RAM
+4E	lwfill03	Fill to keep on long word boundaries
+50	deloadpmm	Deloadable bytes of private memory
+54	deloadsmm	Deloadable bytes of global memory
+58	mpbthresh	Deload threshold for MPB RAM
+5A	lwfill04	Fill to keep on long word boundaries
+5C	pmmthresh	Deload threshold for private memory
+60	smmthresh	Deload threshold for global memory
+64	pmtreq	If 1, task yields on next trap 1 or trap 4, if set
+66	retryflag	Flag to indicate that a retry is in progress
+68	clocktyp	Clock type: if 0, millisecond clock; if 1, real-time clock
+6A	lwfill05	Fill to keep on long word boundaries
+6C	timertcb	Task pointer of timer task
+70	diagflag	Current debug support tools set
+72	lwfill06	Fill to keep on long word boundaries
+74	binclock	Binary time of day, accurate to 0.1 second

(Continued)

Table F-7. System Configuration Table *(Continued)*

Offset	Field Name	Description
+78	decclock	Binary-coded-decimal date/time, accurate to 0.1 second
+80	assumed_year	Assumed year (used by the Executive)
+82	lwfill07	Fill to keep on long word boundaries
+84	firewall	Address of interrupt firewall chain
+88	prilist	Eight entries, one for each task priority level (starting with priority 0); each 16-byte entry is in the form of a QCB, described in appendix H, and indicates information about the tasks ready to execute at that priority
+108	globmem	Global memory extent list; a 16-byte entry in the form of a QCB, described in appendix H
+118	privmem	Private memory extent list; a 16-byte entry in the form of a QCB, described in appendix H
+128	mpbmem	MPB RAM memory extent list; a 16-byte entry in the form of a QCB, described in appendix H
+138	iptlist	Defined interrupts list; a 16-byte entry in the form of a QCB, described in appendix H
+148	lbuffq	Data buffer queue; a 16-byte entry in the form of a QCB, described in appendix H
+158	sbuffq	Descriptor buffer queue; a 16-byte entry in the form of a QCB, described in appendix H
+168	data_buffer_count	Number of data buffers
+16A	descriptor_buffer_count	Number of descriptor buffers
+16C	expire_stp	Expire state transition processor timer
+16E	lwfill08	Fill to keep on long word boundaries
+170	stack_overflow_space	Size of allocated stack overflow area
+174	task_overflowed	Task pointer of task that has overflowed its stack

(Continued)

Table F-7. System Configuration Table *(Continued)*

Offset	Field Name	Description
+178	pc_chkinst_address	Program counter (PC) where CHK instruction executed
+17C	usp_chkinst_address	Value of User Stack Pointer when CHK instruction executed
+180	mpb_light_state	Status of MPB lights
+184	idle_loop_count	Number of executions of idle loop since last cleared
+186	lwfill09	Fill to keep on long word boundaries
+188	reservetop	Largest numerical address in reserved memory
+18C	rsvfree	Number of bytes of reserved RAM memory
+190	rsvfrag	Number of extents of reserved global memory
+192	lwfill10	Fill to keep on long word boundaries
+194	rsvmem	Reserved RAM memory extent list
+1A4	memory_state	0=GOOD, 1=FAIR, 2=POOR, 3=CONGESTED
+1A6	buffer_state	0=GOOD, 1=FAIR, 2=POOR, 3=CONGESTED
+1A8	stp_timer	Timer identifier of state transition processor
+1AC	cio_b	CIO port B bit settings
+1AE	cio_c	CIO port C bit settings
+1B0	supervisor_state_ok	If 1, supervisor state OK; if 0, user task
+1B2	mpb_type	If 0, MPB; if 1, MPB-II; if FF, ICA
+1B4	pmm_allocated_any	Amount of PMM allocated
+1B8	available_to_any	Amount of PMM available
+1BC	current_pmm_required	Current amount of PMM required
+1C0	maximum_pmm_required	Maximum amount of PMM required
+1C4	deload_pmm_any	Deloadable amount of PMM
+1C8	boot_startup_completed	If FALSE, not completed; if TRUE, completed

System Data Record

The system data record contains information useful in dump analysis. Its address may be found using the DISMM subcommand. The system data record is structured according to the system_data_type, which is defined below.

The entire system data record can be displayed using the following Dump Analyzer subcommand:

```
DISM A=SYSTEM_DATA RC=0A2
```

Table F-8 summarizes the fields in the system data record. All offsets are in hexadecimal.

Table F-8. System Data Record

Offset	Field Name	Description
+0	system_name	Length in characters of system name
+22	system_state	The current state of the system; 0000(16) indicates operational
+24	mpb_use	MPB busy percent; average of last ten 10-second periods
+3A	last_deadman_reset	The BCD clock value at time of last deadman time-out reset
+3E	time_system_became_operational	15 four-bit fields indicating system startup date/time
+46	system_address	Network address of the system
+50	master_clock	Pointer to master clock task
+54	default_channel_trunk_defined	Set TRUE if default channel trunk is defined
+55	routing_system	This system forwards routing information.
+56	default_channel_trunk	Channel trunk name

(Continued)

Table F-8. System Data Record *(Continued)*

Offset	Field Name	Description
+78	build_level	Software build level installed
+7A	smd_serial_number	8-character string indicating this system's serial number
+82	abort_system	Set TRUE if KILL_SYSTEM command is being executed
+84	library_version	Version code found within the last accepted help offer PDU for the current load
+86	helping_system_id	System identifier found within the last accepted help offer PDU
+8C	boot_lib_ptr	Pointer to boot library
+90	osi_primary_subnet_state	Primary subnet state type
+92	cdcnet_nsap_address_prefix	Configured CDCNET NSAP address
+9E	title_threshold_cost	Cost of routing title threshold
+A2	start_configuration	If TRUE, configuration started
+A3	validate_user	Network validation defined for this DI
+A4	default_validation_domain	Domain used for network validation login when none specified
+C6	maximum_login_attempt	Retry limit for network validation login

Example:

Following is a section of DI memory that includes the system data record. This display was made from a dump file using the Dump Analyzer's DISM subcommand.

```
STARTING ADDRESS:    11684C

HEX ADDR              HEXADECIMAL DATA                     ASCII DATA

   11684C  0006 4D44 495F 3745 2020 2020 2020 2020     MDI_7E
   11685C  2020 2020 2020 2020 2020 2020 2020 2020
   11686C  2000 0000 0013 0002 000F 0013 0026 000F                 &
   11687C  001A 0002 0019 002C 0006 0005 0D5C 8809            ,    \
   11688C  2713 5421 1940 0000 0001 0800 2510 007E    ' T
   11689C  0000 0000 0100 0005 244D 4349 3720 2020            $MCI7
   1168AC  2020 2020 2020 2020 2020 2020 2020 2020
   1168BC  2020 2020 2020 2000 5504 2343 4443 4E45            U #CDCNE
   1168CC  5423 0000 5514 0000 0000 0000 0010 7430    T#  U          t0
   1168DC  0101 0003 0000 0000 0000 0000 0000 0000
   1168EC  0000
```

Hardware Status Tables

The following hardware status tables are described in this subsection:

- Major card status table
- LIM status table
- Port status table
- SMM bank status table
- PMM bank status table

Major Card Status Table

The major card status table (MCST) is used to record the status of each of the DI's major boards. Each board has its own MCST entry as shown in table F-9. Chapter 9 describes how to locate and interpret the MCST from a DI dump file using the DI Dump Analyzer.

Table F-9. Major Card Status Table Entry

Offset	Field Name	Description
+0	device	Board-identification string
+C	state	Board state; see table 9-3
+E	status	Board status; see table 9-4
+10	slot_number	Major card slot number
+12	card_type	Card type of major card
+14	version	Major card version number
+16	rom_level	ROM level of board in slot
+18	status_extension	Status extension for this board
+1A	board	Board status type
+1E	icb_write_register_zero	Contents of the Internal Control Bus (ICB) write register zero
+20	icb_write_register_one	Contents of ICB write register one
+22	qk_look_diag	Quicklook diagnostics status; if non-zero, testing failed
+24	icb_address	The ICB address for this slot
+28	itb_address	The ITB address for this slot
+2C	table_address	Address of user's configuration table
+30	dst_ptr	Diagnostic status table pointer
+34	pst_address	ICA port table pointer

LIM Status Table

Each LIM board slot in the DI has a LIM status table (LST) entry associated with it that records the associated LIM's status. Chapter 9 describes how to locate and interpret the LST from a DI dump file using the DI Dump Analyzer.

The structure of an LST entry is described in the following table.

Table F-10. LIM Status Table Entry

Offset	Field Name	Description
+0	device	Board-identification string
+C	state	Board state; see table 9-3
+E	status	Board status; see table 9-4
+10	pst_address	PST pointer
+14	lim_number	LIM card slot
+16	board_in_slot	TRUE=LIM board physically in LIM slot
+17	degraded	TRUE=LIM is degraded
+18	cim_sw_8_primary	TRUE=switch is on
+19	cim_sw_10_local	TRUE=switch is on
+1A	lim_kind	Kind or type of LIM board: 0000 RS-449 0001 RS-232 0002 URD_BP1500 0003 URD_B300 0004 URD_E_SERIES 0005 URD_LINE_WRITER 0006 URD_FASTBAND 0007 URD_DATA_PRODUCTS_BASICS 0008 URD_CENTRONICS_360X_720X 0009 URD_CENTRONICS_703 000A V.35 000B X.21 000C LIM_SLOT_EMPTY
+1C	lim_id	Value of ID read from the LIM
+1E	parent_cim_slot	Parent CIM card slot
+20	ports_supported	Number of PORTs on LIM
+22	dst_ptr	Diagnostic status table pointer

Port Status Table

The status of each port associated with a LIM is recorded in its own port status table (PST) entry. Chapter 9 describes how to locate and interpret a PST from a dump file using the DI Dump Analyzer.

Each PST entry has the structure described in the following table.

Table F-11. Port Status Table Entry

Offset	Field Name	Description
+0	device	Port-identification string
+C	state	Port state; see table 9-3
+E	status	Port status; see table 9-4
+10	port_number	Port slot number
+12	port_owner	Port owner: 0000 Port available 0001 HDLC owner 0002 X.25 owner 0003 LCM owner
+14	table_ptr	Address of user configuration table
+18	dst_ptr	Diagnostic status table pointer

SMM Bank Status Table

Each system main memory (SMM) board has an SMM bank status table (SBST) associated with it that records its status. Chapter 9 describes how to locate and interpret an SBST from a DI dump file using the DI Dump Analyzer.

The SBST is structured as described in the following two tables. The main structure is described in table F-12. The bank tables field in this table contains one entry for each SMM memory bank (the number of memory banks is indicated in the first field of the SBST). Each of the bank tables field is structured as described in table F-13.

Table F-12. SMM Bank Table Type

Offset	Field Name	Description
+0	number_of_banks	Number of memory banks
+2	memory_block_size	SMM memory block size
+6	bank tables	See table F-13

Table F-13. Bank Tables Field (SMM Bank Table Type)

Offset	Field Name	Description
+0	device	SMM bank-identification string
+C	state	Bank state; see table 9-3
+E	status	Bank status; see table 9-4
+10	block_address	Start of memory block address
+14	dst_ptr	Diagnostic status table pointer

PMM Bank Status Table

If there is a private memory module (PMM) in the DI, it has a PMM bank status table (PBST) associated with it that records its status. Chapter 9 describes how to locate and interpret a PBST from a DI dump file using the DI Dump Analyzer.

The PBST structure is described in the following table.

Table F-14. PMM Bank Table Type

Offset	Field Name	Description
+0	memory_block_size	PMM memory block size
+4	device	PMM bank-identification string
+10	state	Bank state; see table 9-3
+12	status	Bank status; see table 9-4
+14	dst_ptr	Diagnostic status table pointer

Link Information Block

Link information blocks (LIB) contain link related information needed to provide the services and functions associated with a particular link.

The following example shown in table F-15 features the common part of all LIBs.

Table F-15. Link Interface Block

Offset	Field Name	Description
+0	next_linked_lib	Next LIB on network solution list
+4	nib_ptr	Pointer to owning NIB
+8	output_qcb	Queue control blocks
+48	total_qcb_qcharacters	Total bytes in all queues
+4C	interactive_data_amount	Amount of data to transmit from interactive queue before selecting data from batch priority queue
+50	interactive_bandwidth	Factor of interactive data to be transmitted for every datagram of batch data
+52	batch_bandwidth	Factor of batch data to be transmitted for every datagram of interactive data
+54	output_queue_limit	Configured value for number of bytes in queue before considered congested
+58	previous_queuing_bytes_delay	Delay to transmit message
+5C	network_significant_difference	Significant difference based on speed
+5E	sixty_seconds_of_bytes	Sixty-seconds worth of bytes
+62	current_congested_byte_limit	Current value for number of bytes in queue before considered congested

(Continued)

Table F-15. Link Interface Block *(Continued)*

Offset	Field Name	Description
+66	nxt_lib_ptr	Chain to next LIB on rotary
+6A	link_status	Status of this link
+6C	previous_link_status	Previous link status of this link
+6E	ssr_task_id	SSR task identifier
+72	ssr_tracing	Diagnostic trace
+73	ssr_collecting_stats	Collecting statistics
+74	ssr_sleeping_lock	Intranet LIB lock
+75	ssr_sleeping	SSR needs wake-up call
+76	ssr_waiting_normal_data	Wakeup SSR for normal data
+77	ssr_waiting_priority_data	Wakeup SSR for priority data
+78	lib_defined	LIB defined or booted
+79	congested	Link is congested
+7A	ssr_data_req_proc	Get data
+82	ssr_data_ind_proc	Send data
+8A	ssr_status_ind_proc	Send status
+92	link_type	Owner of LIB
+94	trunk_name	Name of LIB
+B6	lmcb	Link monitor control block
+FC	traffic_type	Network traffic type, must = traffic type on
+FE	last_operational_transition	Used by DISNS for D and T of last transition
+108	why_link_went_down	Last link down reason
+10A	stay_down	Time left for unstable trunk to stay down
+10C	stay_down_period	Unstable trunk stay down period

Timer Queue

The timer queue manages timers associated with system tasks that must be scheduled for timed or periodic execution. The queue control block (QCB) structure for this queue is at entry point TIMQCB and can be displayed with the following Dump Analyzer subcommand.

```
DISM A=TIMQCB BC=0C
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS      10008

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

   10008   0020 107C 0019 66E2 0010 2512                    |  f   %
```

The second four bytes in this display point to the first entry in the timer queue. The third set of four bytes point to the last entry in the queue.

Continuing this example, use the following Dump Analyzer subcommand to display a linked list of all entries in the timer queue.

```
DISLL A=1966E2 LO=0 BC=1C
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS·    1966E2

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

  1966E2   0010 2408 0016 2154 494D 000E 0004 FFB0       $    !TIM
  1966F2   0000 03E8 001E 16A4 001D EF20

ADDRESS OF NEXT ELEMENT·     102408

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

  102408   0010 2538 0016 2154 494D 000E 0005 0014       %8   !TIM
  102418   0000 03E8 0011 7012 0011 4EDE                      p    N

ADDRESS OF NEXT ELEMENT:     102538

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

  102538   0010 27E4 0016 2154 494D 000E 0005 0014       '    !TIM
  102548   0000 0BB8 0010 6B22 0011 847A                      k"   z
                     .      .           .
     .               .            .     .     .     .
     .      .        .                  .     .     .
```

Use table F-16 to interpret information in timer queue entries.

Table F-16. Timer Queue

Offset	Field Name	Description
+0	next_one	Pointer to next timer in queue
+4	length	Length of remaining fields in entry
+6	timer entry identifier	The identifying string '!TIM'
+A	code	Identifying code, as follows: 0E(16) = Periodic timer 0F(16) = Interval timer 10(16) = At-time timer 11(16) = Periodic timer beginning at-time
+C	tod	Time of day to execute timed procedure (in milliseconds)
+10	period	Time of day period, if periodic timer (in milliseconds)
+14	param	Pointer to cell holding subroutine parameter(s)
+18	proc	Address of subroutine

Directory Data Stores

The Directory Management Entity (ME) in each DI maintains title information in the three data stores described in table F-17.

Table F-17. Directory Data Stores

Data Store	Description
Registration data store (RDS)	This data store contains all the currently registered titles. There is no maximum number of entries in this data store. Entries are forward-chained from top to bottom, by the most recent binary clock value.
Translation data store (TDS)	This data store contains a list of the most recent translation data units received from other systems. Entries are ordered top-to-bottom in a least-recently-used chain. When the maximum number of entries is reached (currently 100), the least-recently-used entry is deleted. The structure of this data store is like that of the registration data store.
Translation request data store (TRDS)	This data store contains all the currently active translation requests. Entries are ordered top-to-bottom, in a forward chain, by most most recent binary clock value. There is no maximum number of entries in this data store.

The addresses of these data stores can be found with the following Dump Analyzer subcommand:

```
DISM A=DIR_DS BC=0C RC=3
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    1403AC

HEX ADDR            HEXADECIMAL DATA               ASCII DATA

 1403AC  5452 4453 001C B652 7FFF FFFF            TRDS  6R
 1403B8  5244 5320 0010 8AE6 0005 2616            RDS    f  &
 1403C4  5444 5320 0027 1358 7FFF FFFF            TDS  ' X
```

The address of the RDS is in bytes 10(16) through 13(16) of this display. The address of the TDS is in bytes 1C(16) through 1F(16). The address of the TRDS is in bytes 4 through 7.

The Registration Data Store

The RDS can be displayed with the Dump Analyzer DISPLAY_LINKED_LIST subcommand, using the address found in bytes 10(16) through 13(16) of the display from address DIR_DS. Use a byte count of 75(16) to show each complete RDS entry with the first 26(10) bytes of the associated title.

The following subcommand continues with the example in progress.

```
DISLL A=108AE6 BC=75 LO=4
```

Display from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS     108AE6

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

  108AE6  5244 5320 0021 95A4 0005 2616 A06D 0027   RDS
  108AF6  1358 0000 0000 0800 2510 007E 8809 2713    X      %  ~  '
  108B06  5430 4100 0000 0000 0000 0010 8B42 0011   T0A             B
  108B16  0000 0000 0000 0000 0000 0007 0F00 0000
  108B26  0000 0000 0008 0025 1000 7E01 7374 6163           %  ~ stac
  108B36  6B7E 9B26 7374 6163 0103 C000 2449 5F4C   k~ &stac  @ $I_L
  108B46  4F47 5F4D 455F 4341 5445 4E45 5400 0000   OG_ME_CATENET
  108B56  0000 0000 00

ADDRESS OF NEXT ELEMENT    2195A4

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

  2195A4  5244 5320 0010 6D36 0005 B14C 9070 0027   RDS   m6  1L p '
  2195B4  3ED6 0000 0000 0800 2510 007E 8809 2713   >V      %  ~  '
  2195C4  5304 0130 0003 0000 0000 0021 9600 0013   S  0
  2195D4  0021 9613 0001 0000 0000 0007 0F00 0000
  2195E4  0000 0000 0008 0025 1000 7E01 0000 0000           %  ~
  2195F4  0000 9B22 C90A 001D 0103 C04F 2449 5F41     "I      @O$I_A
  219604  4C41 524D 5F4D 455F 4341 5445 4E45 5431   LARM_ME_CATENET1
  219614  3938 372C 20                              987,

ADDRESS OF NEXT ELEMENT    106D36

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

  106D36  5244 5320 0010 7398 000E 09CB 8074 0000   RDS   s    K t
  106D46  0000 0000 0000 0800 2510 007E 8809 2713           %  ~  '
  106D56  5154 2550 0000 0000 0000 0010 6D92 0018   QT%P           m
  106D66  0000 0000 0000 0000 0000 0007 0F00 0000
  106D76  0000 0000 0008 0025 1000 7E01 6B7E 0101           %  ~ k~
  106D86  7374 03FE 6B7E 0102 0103 8057 2453 5953   st ~k~      W$SYS
  106D96  5445 4D5F 2444 495F 3038 3030 3235 3130   TEM_$DI_08002510
  106DA6  3030 3745 4F                              007E0
     .                  .       .
     .
                        .       .
```

Use table E-18 to interpret the contents of each RDS entry.

Table F-18. Registration Data Store

Offset	Field Name	Description
+0	table_id	4-character ASCII string 'RDS'
+4	link	Pointer to next RDS entry
+8	binclock	Time to broadcast entry; no broadcast if 7FFFFFFF
+C	pdu_sent	Indicates TDU has been sent
+C:1	broadcast_counter	Decrementing counter
+C:4	indication_returned	Not used
+C:5	length	Length of entry, in bytes
+E	key_link	Forward link for same hash of title
+12	dir_id	Directory entry identifier; system address and BCD date and time
+24	refresh_counter	Not used
+26	change_counter	Identifies the number of changes to the title
+28	trds_usage_count	Not used
+2A	title_ptr	Pointer to title string
+30	userinfo_ptr	Pointer to string of user information
+36	password	A password must be supplied to change or delete the title
+3A	address	Address associated with the title
+58	priority	Priority of the title (1..0ff(16))
+59	service	Directly accessible service
+5A	translation_domain	Domain where title may be translated
+5A:1	distribute_title	Boolean; set to distribute the title over translation domain
+5A:2	class	Title class (ordinal: cdna_internal, cdna_external)
+5C	title	The title (maximum length 255 characters)

The Translation Data Store

The TDS can be displayed with the Dump Analyzer DISPLAY_LINKED_LIST subcommand, using the address found in bytes 1C(16) through 1F(16) of the display from address DIR_DS. Use a byte count of 75(16) to show each complete TDS entry with the first 26(10) bytes of the associated title.

The following subcommand continues with the example in progress.

```
DISLL A=271358 BC=75 LO=4
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS    271358

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  271358  5444 5320 0027 3ED6 0003 72FE 006D 0021   TDS  '>V  r~ m
  271368  95A4 0000 0001 0800 2510 0088 8809 2613    $       %      &
  271378  4502 3070 0003 0000 0000 0027 13B4 0011   E 0p          ' 4
  271388  0000 0000 0000 000D 0000 0002 0000 0001
  271398  0800 2510 0088 8205 6B7E 2020 2020 2020     %      k~
  2713A8  2000 0000 0000 0000 0103 8020 2449 5F4C               $I_L
  2713B8  4F47 5F4D 455F 4341 5445 4E45 5420 2020   OG_ME_CATENET
  2713C8  2020 2020 20

ADDRESS OF NEXT ELEMENT.   273ED6

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  273ED6  5444 5320 001E 115E 0003 730F 0070 0000   TDS      ^  s  p
  273EE6  0000 0000 0001 0800 2510 0088 8809 2613            %      &
  273EF6  4405 9420 0003 0000 0000 0027 3F32 0013   D              '?2
  273F06  0027 3F45 0001 0000 0000 0002 0000 0001    '?E
  273F16  0800 2510 0088 81E6 2020 2020 2020 2020     %    f
  273F26  2000 0000 0000 0000 0103 8006 2449 5F41               $I_A
  273F36  4C41 524D 5F4D 455F 4341 5445 4E45 5431   LARM_ME_CATENET1
  273F46  0001 0001 00

ADDRESS OF NEXT ELEMENT:    1E115E

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  1E115E  5444 5320 0000 0000 0004 6024 0067 0000   TDS            g
  1E116E  0000 0000 0000 0800 2510 0088 8809 2613            %      &
  1E117E  4245 9140 0003 0000 0000 001E 11BA 000B   BE @
  1E118E  0000 0000 0000 0000 0000 0002 0000 0001
  1E119E  0800 2510 0088 81E1 3D61 2020 2020 2020     %     a=a
  1E11AE  2000 0000 0000 0000 0103 807E 2449 5F43              ~$I_C
  1E11BE  4C4F 434B 5F4D 4574 6163 6B7E 1000 00E4   LOCK_MEtack~    d
  1E11CE  00A1 0000 00
```

Use table F-19 to interpret the contents of each TDS entry.

Table F-19. Translation Data Store

Offset	Field Name	Description
+0	table_id	4-character ASCII string 'TDS'
+4	link	Pointer to next TDS entry
+8	binclock	Binary time of day TDS entry added to the data store or last used
+C	pdu_sent	Not used
+C:1	broadcast_counter	Not used
+C:4	indication_returned	Not used
+C:5	length	Length of entry, in bytes
+E	key_link	Forward link for same hash of title
+12	dir_id	Directory entry identifier; system address and BCD date and time
+24	refresh_counter	Identifies the need for refreshing cache; 0 if refresh needed
+26	change_counter	Identifies the number of changes to the title
+28	trds_usage_count	Identifies the number of translation requests using the TDS entry
+2A	title_ptr	Pointer to title string
+30	userinfo_ptr	Pointer to string of user information
+36	password	Not used
+3A	address	Address associated with the title
+58	priority	Priority of the title (1..0ff(16)) where 1 is highest
+59	service	Directly accessible service
+5A	translation_domain	Not used
+5A:1	distribute_title	Not used
+5A:2	class	Title class (ordinal: cdna_internal, cdna_external)
+5C	title	The title (maximum length 255 characters)

The Translation Request Data Store

The TRDS can be displayed with the Dump Analyzer DISPLAY_LINKED_LIST subcommand, using the address found in bytes 4 through 7 of the display from address DIR_DS. Use a byte count of 60(16) to show each complete TRDS entry with the first 26(10) bytes of the associated title.

The following subcommand continues with the example in progress.

```
DISLL A=1CB652 BC=60 LO=4
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS     1CB652

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

  1CB652  5452 4453 001C B5BA 7FFF FFFF 8872 0000    TRDS  5:      r
  1CB662  0000 2020 2020 2020 2020 2020 2020 2020
  1CB672  2020 2020 2020 2020 2000 001C B698 002B                 6  +
  1CB682  0024 BC1A 0015 1FF2 0000 0000 7FFF FFFF    $<    r
  1CB692  00C0 0000 0000 2449 5F41 4C41 524D 5F4D    @     $I_ALARM_M
  1CB6A2  455F 4341 5445 4E45 5420 2020 2020 2020    E_CATENET

ADDRESS OF NEXT ELEMENT:     1CB5BA

HEX ADDR                HEXADECIMAL DATA                    ASCII DATA

  1CB5BA  5452 4453 0000 0000 7FFF FFFF 8870 0000    TRDS          p
  1CB5CA  0000 0000 454E 4420 4F46 204C 494E 4520        END OF LINE
  1CB5DA  2020 2020 2020 2020 2000 001C B600 0029                 6  )
  1CB5EA  0025 34EE 0015 0546 0000 0000 7FFF FFFF    %4n   F
  1CB5FA  00C0 0000 0000 2449 5F4C 4F47 5F4D 455F    @     $I_LOG_ME_
  1CB60A  4341 5445 4E45 5420 2020 2020 2020 2020    CATENET
```

Use table F-20 to interpret the contents of each TRDS entry.

Table F-20. Translation Request Data Store

Offset	Field Name	Description
+0	table_id	4-character ASCII string 'TRDS'
+4	link	Pointer to next TRDS entry
+8	binclock	Time to broadcast request; no broadcast if 7FFFFFFF
+C	pdu_sent	Indicates TRDU has been sent
+C:1	broadcast_counter	Decrementing counter
+C:4	indication_returned	Indicates dir_indication returned
+C:5	length	Length of entry, in bytes
+E	wait_parms_ptr	Pointer to sleeping task parameters
+12	request_system_address	Address of the system requesting the translation
+27	version3_request	Request was a Version 3 Directory PDU.
+28	searching_tds	Boolean; set if TDS is being searched to satisfy the request
+2A	title_ptr	Pointer to title string
+30	user_id	User identifier
+34	translation_if	Address of user procedure to return translation
+3C	time	Integer
+40	service	Directly accessible service
+41	search_domain	Directory domain (ordinal: local_system, catenet)
+41:1	recurrent_search	Boolean; set if recurrent search
+41:2	class	Title class (ordinal: cdna_internal, cdna_external)
+41:3	wild_card	Boolean; set if wild-card title translation
+42	non_one_priority_rds	Linked list of non_one_priority RDS entry information
+46	rds_tds_scan	Boolean; set if RDS and TDS tables are currently being searched
+48	title	The title (maximum length 255 characters)

OSI IS-IS Routing Data Store

The OSI IS-IS routing support maintains routing information in the three data stores described in table F-21.

Table F-21. IS-IS Routing Data Stores

Data Store	Description
Least cost routing data store (LCRDS)	This data store is used by IS-IS routing support for determining how to send network protocol data units (NPDUs) to their next intermediate destination.
Local DCN data store (Local DCNDS)	This data store describes the networks that are directly connected to the DI.
Received DCN data store (Received DCNDS)	This data store records information about networks that are not directly connected to the DI, but that are directly connected to other (relay) systems on this network solution. Entries are built from routing information data units (RIDUs).

Least Cost Routing Data Store

The LCRDS is organized into rows, each row pointing to a linked list of the data store entries that apply to a single network. Figure F-1 illustrates the relationship between LCRDS rows and entries.

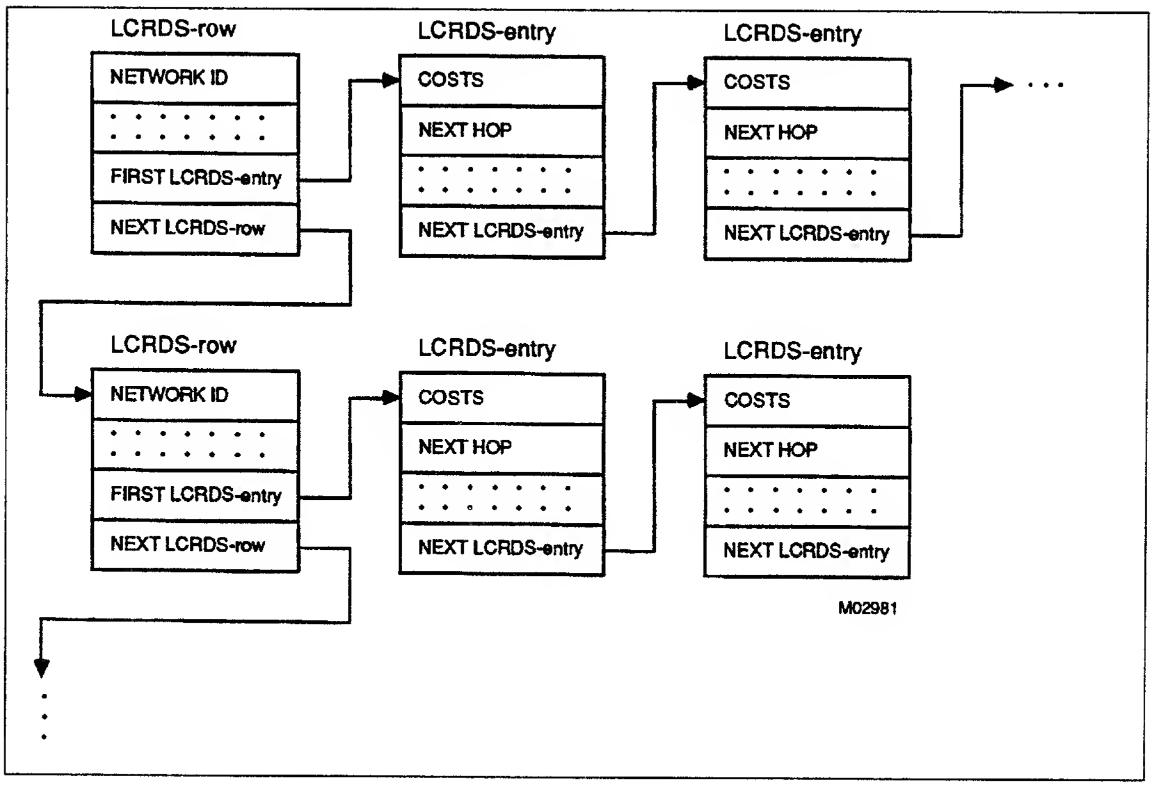


Figure F-1. Least Cost Routing Data Store

A pointer to the first row structure for the least cost routing data store can be displayed with the following Dump Analyzer subcommand:

```
DISM A=II_CURRENT_LCRDS BC=4
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    1340CE

HEX ADDR            HEXADECIMAL DATA                      ASCII DATA

  1340CE  001C 6462                                        db
```

Continuing this example, a linked list of the least cost routing data store *rows* can be displayed with the following Dump Analyzer subcommand:

```
DISLL 1C6462 BC=1A LO=16
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS:   1C6462

HEX ADDR            HEXADECIMAL DATA                      ASCII DATA

  1C6462  0000 0001 0000 0100 0100 0900 25FF FFFF               %
  1C6472  0001 001C BFAC 001C B522                          ?,  5"

ADDRESS OF NEXT ELEMENT     1CB522

HEX ADDR            HEXADECIMAL DATA                      ASCII DATA

  1CB522  0000 0002 0000 0000 0100 0900 25FF FFFF               %
  1CB532  0002 001C C2A4 001C BC8E                          B$  <

ADDRESS OF NEXT ELEMENT     1CBC8E

HEX ADDR            HEXADECIMAL DATA                      ASCII DATA

  1CBC8E  0000 0003 0000 0000 0100 0900 25FF FFFF               %
  1CBC9E  0002 001C C362 001C BF86                          Cb  ?

ADDRESS OF NEXT ELEMENT     1CBFB6
                        .
        .       .
```

Use table F-22 to interpret information in an LCRDS row.

Table F-22. Least Cost Routing Data Store Row

Offset	Field Name	Description
+0	network_id	Network identifier
+4	pseudo_subnet	Boolean; if true, network is a pseudo 180 subnet
+5	changed	Boolean
+6	directly_connected	Boolean
+7	alias_exists	Boolean
+8	multicast	Boolean
+A	broadcast_address	System identifier
+10	valid_lcrds_entry_count	Count of valid LCRDS entries
+12	first_lcrds_entry	Pointer to first least cost routing data store entry
+16	next_row	Pointer to next row

To display the entries associated with a row, enter the following Dump Analyzer subcommand (continuing the example with the first row from the previous display):

```
DISLL 1CBFAC BC=30 LO=2C
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS    1CBFAC

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

  1CBFAC  0000 000A 0000 0000 009D 0001 0000 0001
  1CBFBC  0800 2510 0085 0000 0001 0000 0000 0100     %
  1CBFCC  05D8 0000 0000 0000 0000 0000 0000 0000    X
```

Use table F-23 to interpret information in an LCRDS entry.

Table F-23. Least Cost Routing Data Store Entry

Offset	Field Name	Description
+0	aggregate_cost	Computed routing cost
+4	aggregate_cost_ratio	Ratio used by IS-IS routing support balance traffic load; a negative value indicates that this entry is not valid for routing; 0 indicates this is only valid entry for the associated network
+6	pdu_count	Protocol data unit count
+A	relay_count	Relay count
+C	next_hop_network_id	Next hop network identifier
+10	next_hop_system_id	Next hop system identifier
+16	parent_network_id	Network type
+1A	congested	Boolean
+1B	relay_restricted	Boolean
+1C	unused	Boolean
+1D	obsolete	Boolean
+1E	directly_connected	Boolean
+1F	do_broadcast_in_3b	Boolean
+20	max_pdu_size	Maximum protocol data unit size
+22	congestion_beginning	System address
+2C	next_entry	Pointer to next entry

Local DCN Data Store

To display a pointer to the local DCNDS, use the following Dump Analyzer subcommand:

```
DISM A=II_FIRST_LOCAL_DCNDS_ENTRY BC=4
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS-    12FD8E

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  12FD8E  0026 5662                                      &Vb
```

Continuing the example, a linked list of the local DCNDS entries can be displayed using the following Dump Analyzer subcommand:

```
DISLL 265662 BC=40 LO=32
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS     265662

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  265662  0100 0100 0001 0000 0000 000A 000A 0900
  265672  25FF FFFF 05D8 0000 0001 0800 2510 0085   %    X      %
  265682  000A 0051 0900 25FF FFFF 0000 0001 0000      Q  %
  265692  0000 0000 0000 0000 0000 4A6E 0008 6508              Jn  e
```

Use table F-24 to interpret information in a local DCNDS entry.

Table F-24. Local DCN Data Store Entry

Offset	Field Name	Description
+0	cdna_route_info_nw	Boolean
+1	do_broadcast_in_3b	Boolean
+2	last_send_ridu_ok	Boolean
+3	network_went_down	Boolean
+4	network_type	Network type (HDLC, Ethernet, MCI, X.25, test, pseudo 180 network)
+6	network_status	Network status (up, inactive, up for remote load, terminate)
+8	number_of_ridu_timeouts	Current number of RIDU timeouts on this network
+A	last_cong_related_ridu_count	send_the_ridu .. send_congestion_ridu
+C	last_sent_ridu_cost	Routing cost on last RIDU
+E	configured_cost	Configured cost
+10	cdna_routing_addr	Routing system identifier
+16	max_pdu_size	Maximum protocol data unit size
+18	dcn_entry	DCN definition entry (see table F-26)
+2C	ridus_transmitted	Statistics collection
+34	next_entry	Pointer to next entry

Received DCN Data Store

You can display both a pointer to the first received DCNDS entry and an indication of its length with the following Dump Analyzer subcommand:

```
DISM A=II_RECEIVED_DCNDS_PTR BC=8
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    131CBE

HEX ADDR            HEXADECIMAL DATA                 ASCII DATA

 131CBE  0026 7036 0000 003C                          &p6   <
```

The first four bytes of this display point to the first received DCNDS entry. The second four bytes indicate the byte length of the first entry's array of DCN definitions. These eight bytes are found at an offset of 0A(16) into each DCN data store entry. That is, each DCN data store entry holds a link to the next DCN data store entry in bytes 0A(16) through 12(16).

Continuing the example, a linked list of the first 56(16) bytes of all entries in the received DCN data store can be displayed with the following Dump Analyzer subcommand:

```
DISLL 267036 BC=56 LO=0A
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS:   267036

HEX ADDR            HEXADECIMAL DATA                 ASCII DATA

 267036  0000 004F 001D 5564 0003 001C B30E 0000       O  Ud    3
 267046  0028 0000 0001 0000 0001 0800 2510 0089     (            %
 267056  000A 0011 0900 25FF FFFF 0000 0002 0800         %
 267066  2510 0089 000A 0011 0900 25FF FFFF 0000    %          %
 267076  0003 0800 2510 0089 000A 0011 0900 25FF        %            %
 267086  FFFF 0000 0000

ADDRESS OF NEXT ELEMENT:    1CB30E

HEX ADDR            HEXADECIMAL DATA                 ASCII DATA

 1CB30E  0000 001C 001D 55A1 0002 001C 6488 0000          U
 1CB31E  003C 0000 0001 0000 0001 0800 2510 008C     <            %
 1CB32E  000A 0051 0900 25FF FFFF 0000 3333 0800       Q  %      33
 1CB33E  2510 008C 000A 0051 0900 25FF FFFF 2020    %      Q  %
 1CB34E  2020 2020 2020 1000 0026 F020 0000 0003              &p
 1CB35E  0900 25FF FFFF                               %

ADDRESS OF NEXT ELEMENT:    1C6488
   .        .        .        .        .        .        .
   .        .        .        .        .        .        .
   .        .        .        .        .        .        .
```

Use table F-25 to interpret information in a received DCNDS entry. The number of DCN definitions in a received DCNDS entry is in bytes 8 and 9 of that entry. The link to the next entry is at a byte offset of 0A(16) into the entry, as described above.

Table F-25. Received DCN Data Store Entry

Offset	Field Name	Description
+0	sequence_no	Sequence number; integer
+4	timestamp	Timestamp
+8	dcn_count	Number of DCNs
+A	deleted	True if this entry is deleted
+C	next_entry	Pointer to next entry
+18	dcn_entry	Adaptable array of 14(16)-byte-long DCN definitions (see table F-26)

The array of DCN definitions for any received DCNDS entry can be displayed separately using the Dump Analyzer DISPLAY_MEMORY subcommand with the following parameter values:

- Use the next_entry address found in the preceding received DCNDS entry.
- Use a BYTE_OFFSET of 16(16).
- Use a BYTE_COUNT of 14(16), the length of a single DCN definition.
- Use a REPEAT_COUNT that corresponds with the dcn_count in the received DCNDS entry.

For example, the array of DCN definitions for the second received DCNDS entry of the previous example can be displayed using the following subcommand.

```
DISM 1CB30E BO=16 BC=14 RC=2
```

where 1A1598 is taken from bytes 0A through 0D(16) of the first entry in the preceding display.

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS·    1CB324

HEX ADDR             HEXADECIMAL DATA                    ASCII DATA

  1CB324  0000 0001 0800 2510 008C 000A 0051 0900           %      Q
          25FF FFFF                                    %
  1CB338  0000 3333 0800 2510 008C 000A 0051 0900        33 %      Q
          25FF FFFF                                    %
```

Each DCN definition has the structure described in table F-26.

Table F-26. DCN Definition Entry

Offset	Field Name	Description
+0	sys_address	DCN system address
+A	cost	Routing cost to DCN system
+C	reserved_field	Not currently used
+D	pseudo_subnet	Boolean; if true, network is a pseudo 180 subnet
+D:1	routing_info_changed	Boolean
+D:2	title_info_changed	Boolean
+D:3	network_active	Boolean
+D:4	sap_3a_congestion	If 2, CONGESTED; if 0, UNCONGESTED; 1 not implemented
+D:6	relay_restricted	Boolean
+D:7	case broadcast_network	If true, then broadcast address
+E	broadcast_address	System identifier type

Terminal Support Debug Table

The terminal support debug table is created by the ts_debug program. It is used by a variety of DI software to record and access terminal support information. The terminal support debug table begins with an identifying string, *TS-DEBUG**, and ends with an identifying string, *END-TS-DEBUG*. In between is room for 40 terminal support debug records.

The entire table can be displayed with the following Dump Analyzer subcommand:

```
dism ts_debug rc=81e
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS     1D95BA

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

  1D95BA  2A54 532D 4445 4255 472A 2A20 001D 9C2A   *TS-DEBUG**       *
  1D95CA  4C43 4D20 001E C168 001E F74A 001D 82E2   LCM   Ah  wJ     b
  1D95DA  0021 001E F74A 0000 0000 0006 001E 92F2    !  wJ           r
  1D95EA  4C43 4D20 001E 92F2 001E F796 001D 82E2   LCM    r  w      b
  1D95FA  0021 001E F796 0000 0000 0006 001E 92F2    !  w            r
  1D960A  4C43 4D20 001E C272 001E C9B8 001D 82E2   LCM   Br  I8     b
  1D961A  0021 001E C9B8 0000 0000 0006 001E 92F2    !  I8           r



  1D9D8A  2020 2020 0000 0000 2020 2020 2020 2020
  1D9D9A  2020 2020 2020 2020 2020 2020 0000 0000
  1D9DAA  2020 2020 0000 0000 2020 2020 2020 2020
  1D9DBA  2020 2020 2020 2020 2020 2020 0000 0000
  1D9DCA  2A45 4E44 2D54 532D 4445 4255 472A       *END-TS-DEBUG*
```

Each terminal support debug record is 20(16) bytes long. They begin at address ts_debug + 10. This is a circular table, so after the last 20-byte-long segment of the table is written into, the next entry overwrites the first record in the table. That is, once the table has been filled, the latest record always overwrites the oldest record.

Location ts_debug + 0C contains the address of the last debug record written. From the previous display, this record is at address 1D9C2A. The entry with "**" as the receiver_name is the next debug record to be written into.

Table F-27 describes the fields that are found in each terminal support debug record. All offsets are expressed in hexadecimal.

Table F-27. TSD Debug Record

Offset	Field Name	Description
+0	receiver_name	ASCII name of receiver
+4	sender_task_id	Task identifier of sender
+8	user_info	Bytes of user information
+10	message	Bytes of intertask message
+1C	receiver_task_id	Task identifier of receiver

Another way to display the terminal support debug records from the terminal support debug table is to enter the following Dump Analyzer subcommand:

```
dism ts_debug bo=10 bc=20 rc=40
```

There are 40 blocks of 20 bytes each reserved for terminal support debug records, and these begin ten bytes after the ts_debug entry point.

Display from this subcommand is formatted as follows:

```
STARTING ADDRESS:    1D95CA

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

  1D95CA  4C43 4D20 001E C168 001E F74A 001D 82E2   LCM    Ah   wJ    b
          0021 001E F74A 0000 0000 0006 001E 92F2    '  wJ            r
  1D95EA  4C43 4D20 001E 92F2 001E F796 001D 82E2   LCM      r  w     b
          0021 001E F796 0000 0000 0006 001E 92F2    '  w             r
  1D960A  4C43 4D20 001E C272 001E C9B8 001D 82E2   LCM    Br   I8    b
          0021 001E C9B8 0000 0000 0006 001E 92F2    '  I8            r
            .         .         .         .          .         .
            .                             .          .         .
    .       .         .         .         .          .         .

  1D9D6A  2020 2020 0000 0000 2020 2020 2020 2020
          2020 2020 2020 2020 2020 2020 0000 0000
  1D9D8A  2020 2020 0000 0000 2020 2020 2020 2020
          2020 2020 2020 2020 2020 2020 0000 0000
  1D9DAA  2020 2020 0000 0000 2020 2020 2020 2020
          2020 2020 2020 2020 2020 2020 0000 0000
```

Batch Data Service Debug Table

The batch data service debug table begins with an identifying string, *BDSM_DLOG*, and ends with an identifying string, *END-DEBUG-LOG*. In between is room for 50 batch data service debug records.

The entire table can be displayed with the following Dump Analyzer subcommand:

```
dism bds_log bc=0fc0
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS.   295F30

HEX ADDR              HEXADECIMAL DATA                  ASCII DATA

  295F30  2A42 4453 4D5F 444C 4F47 2A20 0029 64E0   *BDSM_DLOG*  )d
  295F40  5343 4624 5250 4D45 5353 001A F868 0045   SCF$RPMESS  xh E
  295F50  2181 1F53 4D44 5F57 4F52 4B5F 5354 4154   !  SMD_WORK_STAT
  295F60  494F 4E5F 3120 2020 2020 2020 2020 2020   ION_1
  295F70  2020 821F 4C50 3120 2020 2020 2020 2020      LP1
  295F80  2020 2020 2020 2020 2020 2020 2020 2020
  295F90  5343 4624 5249 544D 2020 0000 0000 0001   SCF$RITM
  295FA0  0026 001A F868 0024 2BAA 0000 F868 68A8    &  xh $+*  xhh(
  295FB0  0000 FFFF 001B 68A8 001C D335 0000 FFFF          h(  S5
  295FC0  0404 001B 6852 001C BA0A 001B 0000 0045        hR          E
  295FD0  7374 6163 0029 94D0 0011 43DE 0000 0000   stac ) P  C^
     .



  296E90  5343 4624 5249 544D 2020 0000 0000 0001   SCF$RITM
  296EA0  0026 001A F868 0023 1EFE 0000 F868 68A8    &  xh # ~  xhh(
  296EB0  0000 FFFF 001B 68A8 0025 3A37 0000 FFFF          h( % 7
  296EC0  0404 001B 6852 001C BA0A 001B 0000 0045        hR  -       E
  296ED0  7374 6163 0029 94D0 0011 43DE 0000 0000   stac ) P  C^
  296EE0  2A45 4E44 2D44 4542 5547 2D4C 4F47 2A20   *END-DEBUG-LOG*
```

Each batch data service debug record is 50(16) bytes long. They begin at address bds_log + 10. This is a circular table, so after the last 50-byte-long segment of the table is written into, the next entry overwrites the first record in the table. That is, once the table has been filled, the latest record always overwrites the oldest record.

Location bds_log + 0C contains the address of the last batch data service debug record written. From the previous display, this record is at address 2964E0. The entry with "**" as the receiver_name is the next debug record to be written into.

Table F-28 describes the fields that are found in each batch data service debug record. All offsets are expressed in hexadecimal.

Table F-28. Batch Data Service Debug Record

Offset	Field Name	Description
+0	id	Message identifier
+A	log_cepid	Pointer to connection identifier
+E	log_size	Log message size
+10	log_info	Log message information; string up to 40(16)

Another way to display the batch data service debug records from the batch data service table is to enter the following Dump Analyzer subcommand:

```
dism bds_log bo=10 bc=50 rc=32
```

There are 50 blocks of 50(16) bytes each reserved for batch data service debug records. Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:   295F40

HEX ADDR             HEXADECIMAL DATA                      ASCII DATA

  295F40  5343 4624 5250 4D45 5353 001A F868 0045   SCF$RPMESS  xh E
          2181 1F53 4D44 5F57 4F52 4B5F 5354 4154   !  SMD_WORK_STAT
          494F 4E5F 3120 2020 2020 2020 2020 2020   ION_1
          2020 821F 4C50 3120 2020 2020 2020 2020       LP1
          2020 2020 2020 2020 2020 2020 2020 2020
  295F90  5343 4624 5249 544D 2020 0000 0000 0001   SCF$RITM
          0026 001A F868 0024 2BAA 0000 F868 68A8    &  xh $+*  xhh(
          0000 FFFF 001B 68A8 001C D335 0000 FFFF         h(  S5
          0404 001B 6852 001C BA0A 001B 0000 0045       hR  ·       E
          7374 6163 0029 94D0 0011 43DE 0000 0000   stac ) P  C^
            .                 .           .          .
    .       .                       .                .
    .              .          .           .          .

  296E40  5343 4624 5350 4D45 5353 001A F868 002A   SCF$SPMESS  xh *
          1681 1253 4D44 5F57 4F52 4B5F 5354 4154      SMD_WORK_STAT
          494F 4E5F 3182 0343 5231 0300 0400 0802   ION_1  CR1
          0907 0B00 8C02 0190 0F00 5450 5554 2020              TPUT
          2020 2020 2020 2020 2020 2020 2020 2020
  296E90  5343 4624 5249 544D 2020 0000 0000 0001   SCF$RITM
          0026 001A F868 0023 1EFE 0000 F868 68A8    &  xh # `  xhh(
          0000 FFFF 001B 68A8 0025 3A37 0000 FFFF         h( %:7
          0404 001B 6852 001C BA0A 001B 0000 0045       hR  :       E
          7374 6163 0029 94D0 0011 43DE 0000 0000   stac ) P  C^
```

Batch Gateway Debug Table

The batch gateway debug table is used by DI software to record and access batch gateway information. The batch gateway debug table begins with an identifying string, *DEBUG-LOG*, and ends with an identifying string, *END-DEBUG-LOG*. In between is room for 64 batch gateway debug records.

The entire table can be displayed with the following Dump Analyzer subcommand:

```
dism bgw_log bc=820
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS.   1EAC32

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

  1EAC32  2A44 4542 5547 2D4C 4F47 2A20 001E B102   *DEBUG-LOG*   1
  1EAC42  534C 4349 0002 001C 2522 0000 001D C2AE   SLCI    %"     B.
  1EAC52  002C 0001 FDCC 0010 6DF4 0012 3984 0001    ,  }L  mt  9
  1EAC62  4553 5443 00A1 001C 2522 001D C2AE 0019   ESTC !  %'  B
  1EAC72  FD52 0001 FD6C 0001 0000 0013 0000 0000   }R  }1
  1EAC82  4543 4F4E 0002 0027 DCE8 0000 0000 0019   ECON   '\h
  1EAC92  FD52 0001 FD6C 0001 0000 0013 0000 0000   }R  }1
                      .



  1EB402  4553 5443 00A1 001C 2522 0019 FD3E 0016   ESTC !  %"  }>
  1EB412  A202 0800 2510 0083 0000 0061 0000 0000   "   %      a
  1EB422  4543 4F4E 0014 0027 DCE8 0000 0000 0016   ECON   '\h
  1EB432  A202 0800 2510 0083 0000 0061 0000 0000   "   %      a
  1EB442  2A45 4E44 2D44 4542 5547 2D4C 4F47 2A20   *END-DEBUG-LOG*
```

Each batch gateway debug record is 20(16) bytes long. They begin at address bgw_log + 10. This is a circular table, so after the last 20-byte-long segment of the table is written into, the next entry overwrites the first record in the table. That is, once the table has been filled, the latest record always overwrites the oldest record.

Location bgw_log + 0C contains the address of the last batch gateway debug record written. From the previous display, this record is at address 1EB102. The entry with "**" as the receiver_name is the next debug record to be written into.

Table F-29 describes the fields that are found in each batch gateway debug record. Offsets are expressed in hexadecimal.

Table F-29. Batch Gateway Debug Record

Offset	Field Name	Description
+0	id	Message identifier
+4	log_info	Log message information; string up to 28(16)

The first field in a batch gateway debug record identifies the message, as listed in the following table:

Identifier	Description
BIP	BIP Indication Received
BTSC	BTF(S)/DI Connection Indication Received (via Session Layer)
BTSL	BTF(S)/DI Layer Indication Received (via Session Layer)
ECON	Connection State Table Event
EINP	Input State Table Event
EOUT	Output State Table Event
ERCV	Receiver State Table Event
ESND	Sender State Table Event
ESTC	Status and Control State Table Event
RITM	Intertask Message Received
SLCI	SCF/DI Connection Indication Received (via Session Layer)
SLLI	SCF/DI Layer Indication Received (via Session Layer)
SVCC	SVM Call Confirm Received

If the message identifier is for a state table event (those with identifiers that begin with the letter E), **log_info** contains the following fields:

Field	Length
Event code	16 bits
Control block pointer	32 bits
Event point	32 bits
Secondary event codes	8 bits

The **event** and **secondary event** codes and their meanings vary depending on the state table. The information immediately following these fixed-sized fields depends on the event code. See the following examples:

- mark number
- suppress carriage control flag
- data block clarifier
- accounting data
- a pointer to a file transfer control block

For messages other than state table events (indications, intertask messages, and SVM call confirms), the actual event received gets put into the log.

Another way to display the batch gateway debug records from the batch gateway debug table is to enter the following Dump Analyzer subcommand:

```
dism bgw_log bo=10 bc=20 rc=40
```

There are 64 blocks of 20(16) bytes each reserved for batch gateway debug records. Output from this subcommand is formatted as follows:

```
STARTING ADDRESS      1EAC42

HEX ADDR              HEXADECIMAL DATA                     ASCII DATA

 1EAC42   534C 4349 0002 001C 2522 0000 001D C2AE    SLCI    %"     B
          002C 0001 FDCC 0010 6DF4 0012 3984 0001     ,  }L  mt  9
 1EAC62   4553 5443 00A1 001C 2522 001D C2AE 0019    ESTC !  %"  B
          FD52 0001 FD6C 0001 0000 0013 0000 0000    }R  }1
 1EAC82   4543 4F4E 0002 0027 DCE8 0000 0000 0019    ECON    '\h
          FD52 0001 FD6C 0001 0000 0013 0000 0000    }R  }1


 1EB3E2   534C 4349 0002 001C 2522 0000 0019 FD3E    SLCI    %"     }>
          0018 23AA 0000 0001 FDEA 0001 74BC 0000     #*     }j  t<
 1EB402   4553 5443 00A1 001C 2522 0019 FD3E 0016    ESTC !  %"  }>
          A202 0800 2510 0083 0000 0061 0000 0000    "   %       a
 1EB422   4543 4F4E 0014 0027 DCE8 0000 0000 0016    ECON    '\h
          A202 0800 2510 0083 0000 0061 0000 0000    "   %       a
```

Operator Support Application Table

Mainframe Device Interfaces (MDIs) that run the operator support application (OSA) maintain an operator support table that provides information about network operators who are logged into OSA (for a NOS MDI, this is only true if there is a DEFOS command in the configuration file). The operator support table can be displayed with the following Dump Analyzer subcommand:

```
dism osa_basis rc=0ee
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    1F2200

HEX ADDR              HEXADECIMAL DATA                       ASCII DATA

  1F2200  3330 5F36 3035 6000 0098 0001 FFFF 0900   30_605
  1F2210  25FF FFFF 0800 2510 0086 0059 0C0C 0301   %     %    Y
  1F2220  0401 0000 0001 0000 000A 0025 001F 2230             %   "0
  1F2230  0000 0001 4F50 4552 0000 001C F69A 001F      OPER     v
  1F2240  2242 0000 0000 5850 5254 0000 0000 0000   "B     XPRT
  1F2250  0000 0000 0001 0000 2345 0800 2510 0085            #E  %
  1F2260  8809 2215 4234 2960 0002 0000 0000 0800     " B4)
  1F2270  2510 0085 2BE8 2020 2000 0000 0000 0000   %   +h
  1F2280  0001 001E 6996 0000 0000 2BE8 0000 0000        i      +h
  1F2290  2BE7 0000 0001 001D 3CD0 0000 0000 0000   +g        <P
  1F22A0  0000 0000 0000 0001 B888 0000 0000 7465            8      te
  1F22B0  2061 6E64 2074 696D 6520 6F66 206C 6173    and time of las
  1F22C0  7420 7265 6C6F 4F50 4552 5850 5254 001F   t reloOPERXPRT
  1F22D0  04AE 0000 0000 001F 06FE 0000 0000 001E             ^
  1F22E0  BDBC 0000 0000 001E 0F4E 0000 0000 0000   =<        _N
  1F22F0  0001 2449 5F41 4C41 524D 5F4D               $I_ALARM_M
```

Table F-30 describes the fields in an operator support table.

Table F-30. Operator Support Record

Offset	Field Name	Description
+0	osa_password	Not currently used
+21	log_operator_activity	Boolean
+22	max_active_operators_allowed	Limit on # of operators when buffer or memory is congested; otherwise not used
+26	active_cmds_allowed_per_oper	Not currently used
+2A	last_used_operid	Not currently used
+2C	operator_table_ptr	Pointer to numeric key used to maintain information on active operators
+30	operator_table_root	Root structure of operator tree
+3E	xprttbl_ptr	Pointer to transport connection tree
+42	xport_connection_table_root	Root structure of transport connection tree
+50	alarm_sap_status	Transport status for independent alarm ME SAP
+52	xport_sap_status	Transport SAP status
+54	osa_terminated	Flag indicating CANOS command issued
+55	ind_alarm_me_title_registered	Alarm ME title registered; Boolean
+56	ind_alarm_me_dir_id	Directory identifier of independent alarm ME's title
+68	dir_ind_alarm_transport_ address	Directory transport address record of independent alarm ME
+86	ind_alarm_service_sapid	Transport address of independent alarm ME
+8C	osa_service_sapid	Transport address of OSA
+92	operator_alarm_list_hdr	List of entries describing operators' alarm environment
+A6	connection_mgmt_proc	Transport data traffic procedure address

(Continued)

Table F-30. Operator Support Record *(Continued)*

Offset	Field Name	Description
+AE	max_request_length	Length limit on internet datagram
+B0	operator_table_id	Used to validate operator table
+B4	xport_table_id	Used to validate transport table
+B8	close_all_osa_saps	Procedure to close OSA SAPs when CANOS issued
+C0	display_line	Procedure to display line at local console debugger
+C8	osa_request_if	Procedure to receive command displays from K-display and local console interface
+D0	transmit_cdu	Procedure to transmit commands to dependent command MEs

At an offset of 2C(16) into the operator support table contains the address of the root of a tree used to maintain information about individual operators logged into the operator support application. Each node in this tree points to an operator table that describes a single operator. In the previous display, the tree root address is 1F2230.

You can display the entries associated with this tree with the Dump Analyzer DISPLAY_TREE subcommand. For example, the following subcommand displays the individual operator tables from the previous example:

```
dist 1f2230
```

Output from this subcommand is formatted as follows:

```
Tree Identifier          =   OPER
Number of Nodes in Tree  =   1
Tree Kind                =   numeric

Node    1 of    1 (  284 bytes).  key = 203C

  2512A6  0000 011C 4F50 4552 8809 2215 4234 2920         OPER  " B4)
  2512B6  0000 0000 0000 001F 4620 0000 0000 203C            F      <
  2512C6  1800 0000 5348 5254 4C4F 4B00 0001 001C         SHRTLOK
  2512D6  C8E2 0018 0003 0007 0002 0001 0001 0001    Hb
  2512E6  0005 0001 0009 0000 0000 2020 2020 2020
  2512F6  2020 000B 0000 0000 2020 2020 2020 2020
  251306  000D 0000 0000 2020 2020 2020 2020 000F
  251316  0000 0000 2020 2020 2020 2020 0011 0000
  251326  0000 2020 2020 2020 0001 001C EF54 0015                 oT
  251336  2020 2020 2020 2020 0015 0000 0000 2020
  251346  2020 2020 2020 0017 0000 0000 2020 2020
  251356  2020 2020 0019 0000 0000 2020 2020 2020
  251366  2020 001B 0000 0000 2020 2020 2020 2020
  251376  001D 0000 0000 2020 2020 2020 2020 001F
  251386  0000 0000 002D 0025 1398 0025 13A6 0025           - %   % & %
  251396  13B4 0000 0000 434D 4454 0000 0000 0000      4    CMDT
  2513A6  0000 0001 5243 4E54 0000 001C C84A 0000         RCNT    HJ
  2513B6  0000 414C 4941 0002 0000 0000                 ALIA
```

Table F-31 describes the structure of an operator table, like the one in the previous display.

Table F-31. Operator Table Entry

Offset	Field Name	Description
+0	tree_node_control	Node control structure
+8	login_time	BCD time of operator login
+10	command_line_continued	Boolean
+12	continued_command	Pointer to command continuation
+16	source_address	Command request procedure address
+1E	operator_id	Operator identifier
+20	user_data_ptr	User connection endpoint identifier
+24	operator_user_name	Username of operator
+2C	last_used_cmd_dest	Destinations of last SENC command
+88	last_used_cdu_command_id	Key to correlate the 1..n responses with the nth command
+8A	command_table_ptr	Pointer to command table root
+8E	respcnt_table_ptr	Pointer to response table root
+92	alias_table_ptr	Pointer to alias table root
+96	command_table_root	Command table root
+A4	response_table_root	Response table root
+B4	alias_table_root	Alias table root

Loader Entry Point Table

The loader entry point table is in the form of a tree structure. A pointer to the loader entry point tree root can be found at address 56A(16), in mpb_ram. To display this address using the Dump Analyzer, enter the following subcommand:

```
DISM 56A BC=4
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS       56A

HEX ADDR              HEXADECIMAL DATA                   ASCII DATA

     56A  0010 8674                                       t
```

The loader entry point tree for this dump file can then be displayed with the following command:

```
DIST 108674
```

Output from this subcommand is formatted as follows:

```
Tree Identifier          =   eptb
Number of Nodes in Tree  =   793
Tree Kind                =   string

Node     1 of   793 (    70 bytes)    key = A3CPL_LOG

  1737E6  4133 4350 4C5F 4C4F 4720 2020 2020 2020   A3CPL_LOG
  1737F6  2020 2020 2020 2020 2020 2020 2020 2001
  173806  0016 DE1A 0016 D658 0000 0084 0000 0000      ^   VX
  173816  0017 3850 334C E086 C7CC A450 0008 1000     8P3L GL$P
  173826  0026 0070 0001                             & p

Node     2 of   793 (    70 bytes)    key = A3CPR_RESPONSE

  173858  4133 4350 525F 5245 5350 4F4E 5345 2020   A3CPR_RESPONSE
  173868  2020 2020 2020 2020 2020 2020 2020 2001
  173878  0016 E2AE 0016 D658 0000 0084 0000 0000     b    VX
  173888  0017 38C2 8430 6BB2 BD8E 4410 0008 1000     8B 0k2= D
  173898  0026 0070 0000                             & p

Node     3 of   793 (    70 bytes)    key = A3_PMM_INTERFACE

  1172A6  4133 5F50 4D4D 5F49 4E54 4552 4641 4345   A3_PMM_INTERFACE
  1172B6  2020 2020 2020 2020 2020 2020 2020 2001
  1172C6  0001 697A 0011 7D9A 0000 0042 0000 0000     iz  }     B
  1172D6  0011 7310 2085 C2E6 69CA 0870 0008 1000     s   BfiJ p
  1172E6  0026 0070 0000                             & p
```

Table F-32 describes the structure of a loader entry point, such as those displayed in the format above.

Table F-32. Loader Entry Point

Offset	Field Name	Description
+0	node	Node control
+8	name	Program name
+27	declaration_matching_required	Declaration matching required for this module; Boolean
+28	address	MC68000 address
+2C	module_header_address	Module header pointer
+38	link_address	Link address
+3C	declaration_matching_value	Declaration matching value; string (8)
+44	language	Module language

System Memory Management Table

Table F-33 records information about the state of buffers and system memory in the DI.

Table F-33. System Memory Management Table

Offset	Field Name	Description
+0	percentage_data_buffers	Percent of memory in form of data buffers
+2	stp_period	Integer
+6	total_reserved_memory	Amount of reserved memory
+A	total_alloc_memory	Amount of allocated memory
+E	total_data_buffers	Total number of data buffers
+12	initial_data_buffers	Initial number of data buffers
+16	initial_desc_buffers	Initial number of descriptor buffers
+1A	buffer_percentage	Array of percentages for the four buffer states
+20	memory_percentage	Array of percentages for the four memory states
+26	system_configured	Boolean
+27	change_mm_lock	Boolean; change memory management lock

Tree Root Structure

The tree root structure provides information about a binary tree. Each tree root is 14 bytes long, as described in table F-34.

Table F-34. Tree Root Structure

Offset	Field Name	Description
+0	num_nodes	Total number of nodes in the tree
+4	dump_id	Validity check, should contain user value
+8	type_node	Key type for tree access; (numeric, pointer, string)
+A	link	Pointer to node

Following is an example of a tree root displayed using the DI Dump Analyzer:

```
1A2A74  0000 0001 4F50 4552 0000 001B 3590 001A      OPER    5
```

This tree has just one node, and its key type is numeric.

Line and Terminal Control Blocks G

This appendix documents the following line and terminal control blocks:

- Allocated line control block (ALCB)
- Configured line control block (CLCB)
- Terminal cluster control block (TCCB)
- Terminal device control block (TDCB)
- Data connection control block (DCCB)
- Batch device control block (BDCB)
- Batch output connection control block (BOCCB)
- Batch input connection control block (BICCB)
- Batch input/output station control block (IOSCB)
- SCFS connection control block (SCCB)
- TIP interface record table (TIRT)
- Printer terminal model record

Chapter 9 describes how to locate many of these control blocks in a DI dump file using the DI Dump Analyzer.

Allocated Line Control Block

Figure G-1 shows the general structure of the allocated line control block (ALCB). Table G-1 describes the fields in the ALCB.

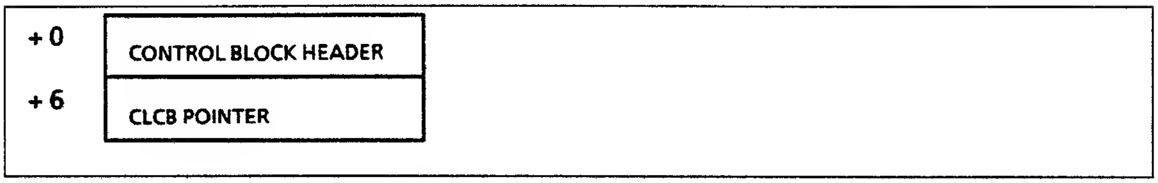


Figure G-1. ALCB Record

Table G-1. Allocated Line Control Block (ALCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	clcb_pointer	Pointer to first CLCB

Configured Line Control Block

Table G-2 describes the fields in the CLCB.

Table G-2. Configured Line Control Block (CLCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	tccb_pointer	Pointer to first-owned TCCB
+A	alcb_pointer	Pointer to owning ALCB
+E	clcb_pointer	Pointer to next CLCB
+12	tip_extension_pointer	Pointer to optional TIP extension
+16	dvmid	Pointer to device manager
+1A	ln	Line name
+1C	lim_adr	LIM and port number
+1E	tbs	Transmission block size
+20	tdp	Terminal definition procedure name
+22	tup	Terminal user procedure name
+24	lst	Line subtype
+26	ls	Defined line-speed
+28	tt	Defined TIP type
+29	lt	Line type (switched, dedicated)
+2A	ft	Framing type (async, sync, sdlc)
+2B	ct	Carrier type (constant, controlled)
+2C	ar	Type of (async) auto recognition
+2D	cct	Connection connect time-out (4-second units)
+2E	cdt	Connection disconnect time-out
+2F	ucl	User connection limit
+30	c	Clocking for LIM
+31	dp	Initial data parity
+32	efc	EIA flow control (RTS/CTS); Boolean
+32:1	pseudo	Pseudo line (for example, X.25 PAD)
+32:2	vu	Validate user
+32:3	vus	Validate user specified
+34	lcsm_task_id	Task identifier of LCSM
+38	control_task_id	Task identifier of controlling task
+3C	tip_task_id	Task identifier of TIP
+40	conf_cmd_queue_ptr	Configuration command queue pointer
+44	tirt_ptr	Address of TIRT entry
+48	a@d	Case 0, 2 Byte word 0..0ffff(16); Case 1, upper 1 Byte indicates # of times a delete_cb sent to TIP, lower 1 Byte indicates # of add_cbs sent to TIP
+4A	activity_count	Number of $io connections and batch activity count

(Continued)

Table G-2. Configured Line Control Block (CLCB) *(Continued)*

Offset	Field Name	Description
+4C	line_speed	Auto-recognized line speed
+4E	state	LCM state-of-the-line
+4F	down_reason	Line down reason code
+50	tip_type	Auto-recognized TIP type
+51	code_set	Auto-recognized code set
+52	parity_type	Auto-recognized parity
+53	connect_timer	Line (dis)connect timer
+54	auto_tup_timer	TUP re-execution timer
+55	line_re_enabled_ timer	Line re_enabled timer (4 second units)
+56	line_re_enabled_ count	Number of times line re_enabled since timer was set
+57	line_enabled	Line is enabled by LCSM; Boolean
+57:1	line_init	Line is initialized by TIP; Boolean
+57:2	defios_done	At least one DEFIOS executed; Boolean
+57:3	tip_switch	Switching from asynch to X.PC TIP; Boolean
+58	state	LCSM state (valid if LCSM active)
+59	ar_state	LCSM auto-recognition state
+5A	itm_part	Timer_itm_rec field of intertask message
+60	sys_task	Pointer to task using timer_services
+64	sys_ptr	System timer pointer
+68	sds_buf1_ptr	User-defined collection buffer 1
+6C	sds_buf2_ptr	User-defined collection buffer 2
+70	group	Statistics group type
+72	log_msg_number	Log message number
+74	log_template_id	Template identifier type
+76	function_proc	Statistics function procedure
+7E	display_proc	Display procedure
+86	next_header	Pointer to sds_header
+8A	collecting	Collecting statistics and next reporting; Boolean
+8B	collecting_buf1	TRUE=buffer1, FALSE=buffer2
+8C	display_template_id	Template identifier to use for display
+8E	input_characters	Characters received
+92	input_blocks	Blocks received
+96	output_characters	Characters sent
+9A	output_blocks	Blocks sent
+9E	input_errors	Input blocks in error
+A0	output_errors	Output blocks in error
+A2	time_outs	Number of time-outs
+BA	stats_sap_id	Statistics SAP identifier

Terminal Cluster Control Block

Table G-3 describes the fields in the TCCB.

Table G-3. Terminal Cluster Control Block (TCCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	tdcb_ptr	Pointer to first-owned TDCB
+A	clcb_ptr	Pointer to owning CLCB
+E	tccb_ptr	Pointer to next TCCB
+12	tip_eptr	Pointer to optional TIP extension
+16	ca	Terminal cluster address
+18	protocol_mode	For example, DCE, DTE, 4A, 4C, 2780, 3780
+1A	connect_time	Connect time
+1E	output_characters	Characters sent
+22	output_blocks	Blocks sent
+26	input_characters	Characters received
+2A	input_blocks	Blocks received
+2E	procs_done	Number of procedures done
+30	xpc_active	XPC active for ATAP
+32	tip_aptr	ATAP TIP accounting data pointer

Terminal Device Control Block

Tables G-4 through G-8 describe the fields in the TDCB.

Table G-4. Terminal Device Control Block (TDCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block; Boolean
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	dccb_pointer	Pointer to first-owned DCCB
+A	tccb_pointer	Pointer to owning TCCB
+E	tdcb_pointer	Pointer to next TDCB
+12	tip_extension_ptr	Pointer to optional TIP extension
+16	device_inactive	Device inactive; Boolean
+16:1	device_stopped	Device stopped by operator; Boolean
+16:2	device_not_ready	Device not ready; Boolean
+16:3	device_down	Device down; Boolean
+16:4	input_flow_control	Input flow control used; Boolean
+16:5	output_flow_control	Ouptut flow control used; Boolean
+18	batch_x25_peer_ptr	Pointer to X.25 peer (batch)
+1C	da	Device address
+1E	dt	Device type
+20	dn	Device name
+22	tup	Terminal user procedure file name
+24	tbs	Transmission block size of device
+26	vu	Validate user
+26:1	vus	Validate user specified
+28	partial_cmd_ptr	Partial command/Control Data
+2C	wc_dccb_ptr	Pointer to DCCB of current working connection
+30	cr_dccb_ptr	Pointer to DCCB of $command_$response dccb
+34	break_time	Time of last break
+38	nr_connects	Number of $input/$ouput connections
+39	do_nesting	Procedure nesting level
+3A	command_q_count	Number of commands in queue
+3B	conn_pending	Connections pending
+3C	ios_operator_device	Required operator device; Boolean
+3C:1	initial_tup	Initial DEFTD TUP executing; Boolean
+3C:2	auto_tup	Re-execute TUP if no $i_o connection; Boolean
+3C:3	defuios_done	At least one DEFUIOS executed; Boolean
+3C:4	device_down	Device down (lcm_devd called); Boolean
+3E	output_queued_itm	ITM on first output queued; Boolean
+3E:1	hold_page	Set by TIP, reset by TDSM; Boolean
+40	connect_time	Connect time
+44	output_characters	Characters sent
+48	output_blocks	Blocks sent
+4C	input_characters	Characters received
+50	input_blocks	Blocks received
+54	procs_done	Number of procedures done

Table G-5. TDCB, Case CPT_VTP

Offset	Field Name	Description
+56	user_validated_or_not_required	User validated or validation not required
+57	validation_retry_limit_exceeded	User exceeded number of validation attempts
+58	user_validation_state	State of network validation
+5A	val_rec_ptr	Pointer to associated validation record
+5E	domain	Network validation domain
+60	username	User name entered
+62	waiting_task	Task waiting for validation completion
+66	retry_count	Number of validation attempts
+68	delete_template_id	Line deletion message
+6C	tm	Terminal model
+6E	eos	End output sequence
+73	crs	Carriage-return output sequence
+76	lfs	Line-feed output sequence
+77	ffs	Form-feed sequence
+82	crd	Carriage-return delay (millisecond units)
+84	lfd	Line-feed delay (millisecond units)
+86	ffd	Form-feed delay (millisecond units)
+88	pl	Device page length
+89	pw	Device page width
+8A	bw	Backspace window
+8B	ncc	Network control character
+8C	blc	Beginning-of-line character
+8D	epc	End-of-partial character
+8E	elc	End-of-line character
+8F	bc	Backspace character
+90	clc	Cancel character
+91	ac	Attention character
+92	elp	CP after ELC (no, cr, lf, cl)
+93	epp	CP after EPC
+94	cs	Code set
+95	p	Parity (zero, mark, even, odd, none)
+96	sa	Status action
+97	ra	Response action
+98	hp	Hold page; Boolean
+98:1	hpo	Hold page (OVER); Boolean
+98:2	fl	Fold line; Boolean
+98:3	e	Echoplex; Boolean
+98:4	cfc	Character flow control (X-ON/X-OFF); Boolean
+99:4	epa	End-of-page action (none, send FF sequence)
+9A	cs_name	Code name set
+9C	xlate_tbl	Code translation table
+A0	alt_xlate_tbl	Alternate code translation
+A4	xlate_mask	Code translation mask
+A5	alt_xlate_mask	Alternate code translation mask
+A6	ccr	Control character replacement
+AA	fkc_name	Function key class name
+AC	fkc_ptr	Pointer to fkc record
+B0	ios	Batch I/O station name

Table G-6. TDCB, Case CPT_BTP

Offset	Field Name	Description
+6C	pptm	Pointer-to-printer terminal model (see table G-22)
+70	mfs	Maximum file size
+74	tbs	Transmission block size of device
+76	pma	Printer-message action
+77	cca	Carriage-control action
+78	pl	Page length
+79	pw	Page width
+7A	CASE batch_usage	See tables G-7 and G-8
+92	scc	Suppress carriage control; Boolean

Table G-7. TDCB/CPT_BTP, Case BU_DEVICE

Offset	Field Name	Description
+7B	cs	Code set
+7C	fs	Form size
+7D	specified_fs	Specified form size
+7E	pd	Print density
+7F	udfa	Undefined format effector action
+80	usfa	Unsupported format effector action
+81	dp	Data parity
+84	vfui_ptr	Pointer to VFU load image
+88	cs_name	Code set name
+8A	ccr	Control character replacement
+8C	xlate_tbl	Code translation table
+90	xlate_mask	Code translation mask
+91	cfc	Character flow control; Boolean
+91:1	o26	Default O26 (card reader only); Boolean
+91:2	uvfu	User changeable VFU; Boolean
+91:3	vpd_changeable	User changeable VPD

Table G-8. TDCB/CPT_BTP, Case BU_STREAM

Offset	Field Name	Description
+7C	spc	Cards/lines to discard
+7E	s	Batch stream auto start; Boolean
+7F	tm	Process data as transparent; Boolean

Data Connection Control Block

Tables G-9 through G-11 describe the fields in the DCCB.

Table G-9. Data Connection Control Block (DCCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	output_queue_ptr	Pointer to first-owned output queue
+A	tdcb_ptr	Pointer to owning TDCB
+E	dccb_ptr	Pointer to next DCCB
+12	tip_eptr	Pointer to optional TIP extension
+16	cn	Name of connection
+18	sn	Name of selected service
+1A	ctype	Type of connection
+1B	ptype	Type of protocol
+1C	lower_layer_id	Session layer connection identifier
+22	cb_qualifier	Unique qualifier
+24	sub	If ctype = ct_$command_$response: +24: status_q_ptr: Status output queue +26: status_q_count: Messages in queue +28: banner_sent: CDCNET banner sent; Boolean If ctype = ct_$input_$output: +24: destination_adr: Destination SAP address
+30	partial_input_ptr	Partial input data pointer
+34	delete_reason	Reason for deleting DCCB
+36	sl_delete_reason	Template identifier received on session clear
+38	owner_task_id	Task_id of owner task
+3C	io_task_id	I/O processor task_id (TIP)
+40	connection_state	State of peer connection
+41	output_q_count	Number of messages in output queue
+42	input_ovf_count	Input passed over IFC limit
+43	connection_number	Connection number
+44	queue_put_open	Output queue open for puts; Boolean
+44:1	queue_get_open	Output queue open for gets; Boolean
+44:2	queue_on_hold	Output queue on hold; Boolean
+44:3	ifc_active	Input flow control active (transport); Boolean
+44:4	ofc_active	Output flow control active (transport); Boolean
+44:5	ofc_end_active	End output flow control ITM send; Boolean
+44:6	expedited_fc_active	Expedited flow control active (transport); Boolean
+44:7	user_int_active	User interrupt in progress; Boolean
+45	partial_input	Partial input sent upline; Boolean
+45:1	ios_operator	I/O station operator connection; Boolean
+45:2	inp_sync	Synchronous on input; Boolean

(Continued)

Table G-9. Data Connection Control Block (DCCB) *(Continued)*

Offset	Field Name	Description
+45:3	out_sync	Synchronous on output; Boolean
+45:4	if_cancel	If cancel character last character of input; Boolean
+45:5	marked_output	Marked output ITM queued
+46	cmd_timer	Time of the last terminal user command execution
+4A	cmd_count	Count of terminal user commands to execute
+4C	input_solicited	Set by TIP, reset by TDSM; Boolean
+4C:1	suppress_elp	TIP suppressing ELP for connection; Boolean
+4C:2	suppress_e	TIP suppressing echo for connection; Boolean

Table G-10. DCCB, Case CPT_VTP

Offset	Field Name	Description
+4E	ibs	Input block size
+50	tml	Transparent message length
+52	tfc	Transparent forwarding characters
+57	ttc	Transparent termination characters
+5C	tfm	Transparent forwarding mask
+7C	iom	Input/output mode
+7D	iem	Input editing mode
+7E	tti	Transparent character time-out interval
+7F	aca	Attention character action
+7F:4	bka	Break key action
+80	tcm	Transparent character mode: (None, Terminate, Forward, Forward/Terminate)
+80:2	tlm	Transparent length mode
+80:4	ttm	Transparent time-out mode
+80:6	tpm	Transparent protocol mode
+81	pcf	Partial character forwarding; Boolean
+81:1	snd	Store NULs and DELs; Boolean
+81:2	sbc	Store backspace character; Boolean
+81:3	ee: boolean	Echo enable
+81:4	ifce: boolean	Input flow control enable
+81:5	ofce: boolean	Output flow control enable
+81:6	pe: boolean	Parity enable
+81:7	ace: boolean	Attention character enable

Table G-11. DCCB, Case CPT_BTP

Offset	Field Name	Description
+4E	bdcb_ptr	Pointer to BDCB
+52	bccb_ptr	Pointer to BOCCB or BICCB
+56	input_q_ptr	Pointer to input queue
+60	partial_input_file_id	Pointer to partial input from output device
+61	current_file_id	Current file identifier
+62	input_q_count	Number of messages in input queue
+63	btbs	Batch transfer block size
+65	data_state	Batch data transfer state
+66	abort_status	Transfer abort status
+67	etpr_sent	ETPR sent to peer application; Boolean
+67:1	internal_disconnect	Internal disconnect (v.s. line); Boolean
+67:2	device_logout	Required operator; Boolean
+67:3	device_stopped	Device stopped by operator; Boolean
+67:4	device_stopped_eoi	Device stopped by operator at next EOI; Boolean
+67:5	device_not_ready	Device not ready; Boolean
+67:6	transparent_data	Batch data is transparent; Boolean
+67:7	discarding_until_ next_cr	Discard input from output service: Boolean

Batch Device Control Block

Tables G-12 through G-14 describe the fields in the BDCB.

Table G-12. Batch Device Control Block (BDCB)

Offset	Field Name	Description
+0	add_req_to_tip	Requested TIP to add; Boolean
+0:1	to_be_deleted	Need to delete control block
+0:2	delete_req_to_tip	Requested TIP to delete; Boolean
+0:3	tip_type	Owning TIP type (debug use)
+1	cb_type	Type of control block
+2	cb_name	ASCII name of control block (debug use)
+6	ioscb_ptr	Pointer to IOSCB
+A	bdcb_ptr	Pointer to next BDCB
+E	tdcb_ptr	Pointer to TDCB
+12	dn	Device name
+14	chabda_response_ptr	Pointer to CHABDA response
+18	state	Device state
+19	dt	Device type
+1A	file_status	File status
+1B	signon_status	Remote system sign-on status
+1C	device_down	Device temporarily down; Boolean
+1C:1	device_stopped	Device disabled; Boolean
+1C:2	device_not_ready	Device not ready; Boolean
+1C:3	device_loading_vfu	Device VFU being loaded; Boolean
+1C:4	device_loading_fpp	File prefix proc being loaded; Boolean
+1C:5	device_loading_ip	Initialization procedure being loaded; Boolean
+1C:6	default_vlp_load_ error	Default VFU not loadable; Boolean
+1C:7	down_fpp_load_error	File prefix proc not loadable; Boolean
+1D	down_ip_load_error	Initialization proc not loadable; Boolean
+1D:1	last_tip_error_type	Type of tlp last reported not loadable
+1D:4	device_available	Device available to host; Boolean
+1D:5	signon_status_ indication	Remote system sign-on indication; Boolean
+1E	pptm	Pointer-to-printer terminal model
+22	mfs	Maximum file size
+26	tbs	Transmission block size of device
+28	pma	Printer-message action
+29	cca	Carriage-control action
+2A	pl	Page length
+2B	pw	Page width
+2C	CASE	See tables G-13 and G-14
+44	fc1	Forms code
+46	fc2	Forms code
+48	fc3	Forms code
+4A	fc4	Forms code
+4C	ec1	External device characteristics
+4E	ec2	External device characteristics

(Continued)

Table G-12. Batch Device Control Block (BDCB) *(Continued)*

Offset	Field Name	Description
+50	ec3	External device characteristics
+52	ec4	External device characteristics
+54	tm	Terminal model
+56	da	Device aliases
+5C	dvlp	Default VFU load procedure
+5E	dvpd	Default vertical print-density
+5F	vpd	Vertical print-density selection
+60	vfus	VFU status
+61	bpc	Number of banner pages
+62	bhf	Banner highlight field
+63	vfut	Type of VFU load being executed
+64	trailer_page	Trailer page to be printed; Boolean
+64:1	explicitly_specified	Trailer page explicitly specified; Boolean
+66	scc	Suppress carriage-control; Boolean

Table G-13. BDCB, Case BU_DEVICE

Offset	Field Name	Description
+2D	cs	Code set
+2E	fs	Form size
+2F	specified_fs	Forms size specified on DEFBD or CHABD
+30	pd	Print density
+31	udfa	Undefined format effector action
+32	usfa	Unsupported format effector action
+33	dp	Data parity
+34	vfui_ptr	Pointer to VFU load image
+38	cs_name	Code name set
+3A	ccr	Control character replacement
+3E	xlate_tbl	Code translation table
+42	xlate_mask	Code translation mask
+43	cfc	Character flow control (ATAP only)
+43:1	o26	Default O26 (card reader only)
+43:2	uvfu	User-changeable VFU
+43:3	vpd_changeable	User-changeable vpd

Table G-14. BDCB, Case BU_STREAM

Offset	Field Name	Description
+2E	spc	Cards/lines to discard
+30	s	Batch stream auto start; Boolean
+31	tm	Process data as transparent; Boolean

Batch Output Connection Control Block

Table G-15 describes the fields in the BOCCB.

Table G-15. Batch Output Connection Control Block (BOCCB)

Offset	Field Name	Description
+0	cb_name	ASCII name BOCB
+4	dccb_ptr	DCCB pointer
+8	data_ptr	Output banner/file position parameter pointer
+C	timer	Abort transfer time-out timer
+E	vlp	VFU load procedure name
+10	state	Output transfer states
+11	abort	Reason for aborting transfer
+12	file_size	File size
+16	file_page_width	File page width
+1A	old_byte_ordinal	Last byte ordinal for amount transferred status
+1E	byte_ordinal	Current file position in bytes
+22	record_ordinal	Current file position in unit records
+26	acc_limit	Accounting limit
+2A	bytes	Accumulated accounting data; # of bytes
+2E	records	Accumulated accounting data; # of records
+32	system_id	Accumulated accounting data; system_id
+34	user_name	Accumulated accounting data; user_name
+36	user_family	Accumulated accounting data; user_family
+38	markack_fac	TRUE=mark acknowledgement facility selected
+38:1	compression_fac	TRUE=compression facility selected
+38:2	tr	TRUE=transparent file, FALSE = ASCII file
+38:3	cmp	TRUE if file in compressed format
+38:4	hold	Reason for suspending transfer
+39	pd	Print density
+3A	mws	Mark acknowledgement window size
+3B	mro	Number of checkmarks awaiting TIP response
+3C	lmr	Last mark received
+3E	lma	Last mark acknowledged
+40	timeout	Transfer time-out interval in seconds
+42	activity_timer	Activity timer
+46	peer_abort_status	Peer abort status code
+47	user_file_name	User file name
+66	sys_file_name	System file name

Batch Input Connection Control Block

Table G-16 describes the fields in the BICCB.

Table G-16. Batch Input Connection Control Block (BICCB)

Offset	Field Name	Description
+0	cb_name	ASCII name 'BICB'
+4	dccb_ptr	DCCB pointer
+8	dir_trid	Translation request identifier (^cell)
+C	actual_destination	Actual destination
+E	requested_destination	Requested destination family
+10	jod	Job output destination
+12	joun	Job output user
+14	jouf	Job output family
+16	bytes_transferred	Bytes sent to peer
+1A	dir_title_ptr	Pointer to directory title
+20	system	Directory entry identifier; system_address
+2A	decclock	Directory entry identifier; bcd_time
+32	state	Batch input transfer states
+34	bytes	Accumulated accounting data; # of bytes
+38	records	Accumulated accounting data; # of records
+3C	system_id	Accumulated accounting data; system_id
+3E	user_name	Accumulated accounting data; user_name
+40	user_family	Accumulated accounting data; user_family
+42	activity_timer	Activity timer
+46	tr	TRUE=transparent file, FALSE = ASCII file
+46:1	abort	Status code for data transfer phase commands
+47:1	peer_abort_status	Peer abort status code
+49	user_job_name	User job name
+68	system_job_name	System job name

Batch Input/Output Station Control Block

Tables G-17 through G-19 describe the fields in the IOSCB.

Table G-17. Input/output Station Control Block (IOSCB)

Offset	Field Name	Description
+0	cb_header	Control block header
+6	ioscb_ptr	Pointer to next IOSCB
+A	sccb_ptr	Pointer to SCCB
+E	bdcb_ptr	Pointer to first BDCB
+12	tdcb_ptr	Pointer to console TDCB for user ios
+16	timer_id	Timer request identifier
+1A	canios_taskid	Task identifier of CANIOS command processor
+1E	state	Station state
+20	user_io_station	Station defined by DEFUIOS command
+21	predefined_io_station	0=dynamic, 1=predefined
+22	operator_login	0=not logged in, 1=logged in
+23	check_ios_unique	Operator loggin required
+24	connection_type	0=180, 1=170
+26	c170_bgw_address	C170 batch gateway address
+44	user_name	Name of login private IOS user
+46	user_family	User family of private IOS user
+48	c180_control_facility	C180 control facility name
+4A	io_station_name	I/O station name (DEFIOS_REC)
+4C	control_facility	Control facility name
+4E	default_destination	Default input file destination
+50	store_forward_ destination	Job input, if requested, not available
+52	destination_ unavailable_action	0=stop, 1=drop
+54	station_usage	CASE su_public, su_private, or su_ntf. See tables G-18 and G-19.

Table G-18. IOSCB, Case SU_PUBLIC, SU_PRIVATE

Offset	Field Name	Description
+54	required_operator_device	Device name of controlling console
+56	io_station_alias	Alias I/O station name
+6C	pm_action	PM message action
+6D	file_acknowledgement	File ACK on or off
+6D:1	route_job_command_required	Route card required option

Table G-19. IOSCB, Case SU_NTF

Offset	Field Name	Description
+54	default_file_destination	Default file destination
+56	arscb_ptr	Pointer to first ARSCB
+5A	next_add_accessible	Pointer to next ARSCB to be configured
+5E	line_name	Name identifier record for line
+60	terminal_user_procedure	Pointer to TUP
+62	line_speed	Line speed
+64	logical_line_number	Logical line number
+66	inactivity_timer	Inactivity timer
+68	authority_level	Remote system authority level
+69	wait_a_bit	Wait flag
+6A	positive_ack	Positive acknowledgement
+6B	remote_system_protocol	BSC protocol type
+6C	local_system_name	Local system name
+6E	route_back_position	Position to insert route back
+6F	remote_system_type	Remote system type
+6F:4	request_permission_retry	Resend Transmission permission

SCFS Connection Control Block

Table G-20 describes the fields in the SCCB.

Table G-20. SCFS Connection Control Block (SCCB)

Offset	Field Name	Description
+0	cb_header	Control block header
+6	sccb_ptr	Pointer to next SCCB
+A	ioscb_ptr	Pointer to first IOSCB
+E	cepid	Connection endpoint identifier
+14	cb_qualifier	Control block unique qualifier
+16	state	Peer connection state
+18	control_facility	Control facility name
+1A	timer_id	Reconnect timer identifier
+1E	sap_address	SAP of control facility
+3C	connection_type	0=180, 1=170
+3E	dir_priority	Translate priority
+40	dir_tid	Save directory translate identifier
+44	system	Directory identifier record; system_addr
+4E	decclock	Directory identifier record; bcd_time
+56	dir_title_ptr	Pointer to directory title
+5C	flow_control_active	Session flow control on SCF; Boolean

TIP Interface Record Table

Table G-21 describes the fields in the TIRT.

Table G-21. TIP Interface Record Table (TIRT)

Offset	Field Name	Description
+0	name	TIP name
+C	min_name_length	Minimum name length
+E	default_tbs	Default TBS
+10	default_ft	Default FT
+12	default_lcs	Default LCS
+14	min_line_speed	Minimum line-speed
+16	max_line_speed	Maximum line-speed
+18	tn	ASCII name of the TIP
+1A	tup	Default terminal user procedure name
+1C	tbs	Default transmission block size
+1E	ca	Default cluster address
+20	da	Default device address
+22	ft	Default framing type for TIP
+23	lcs	Level of line-control support
+24	vu	Validate user
+26	tip_defined	DEFT command processed
+26:1	tip_load_state	State of TIP loading
+27	nr_active_tasks	Number of active TIP line tasks
+28	single_tip_taskid	Single TIP task identifier
+2C	start_adr	TIP entry address for line task
+34	sds_buf1_ptr	First user-defined collection buffer
+38	sds_buf2_ptr	Second user-defined collection buffer
+3C	group	Log group
+3E	log_msg_number	Log message number
+40	log_template_id	Log template identifier
+42	function_proc	Statistics function procedure
+4A	display_proc	Display procedure
+52	next_header	Pointer to next SDS header
+56	collecting	Collecting statistics & next reporting; Boolean
+57	collecting_buf1	TRUE=buffer1, FALSE=buffer2
+58	display_template_id	Template identifier used for display
+5A	stats_sapid	Statistics SAP identifier
+5C	input	TIP input transmission block statistics
+6C	output	TIP output transmission block statistics
+7C	validation_success	Successful network validation login attempts
+80	validation_failures	Unsuccessful network validation login attempts
+AC	stack_size	Stack size in 38 byte units
+AD	max_nr_commands	Maximum commands on LCM queue
+AE	stack_residence	Preferred stack residence
+AE:6	delc$net_action	Delc net action
+AF	task_need	Task requirement for the TIP
+B0	control_task_id	Controlling task (for example ATAP)

(Continued)

Table G-21. TIP Interface Record Table (TIRT) *(Continued)*

Offset	Field Name	Description
+B4	an@av_default_ptr	Default values for an/av conn/term/batch attributes
+C0	an@av_validation_ptr	An/av validation array for conn/term/batch attributes
+CC	dedicated_delay	Delay (seconds) after dedicated line down
+CE	disca_set	DISCA displayed attributes (default all)
+D2	dista_set	DISTA displayed attributes (default all)
+D6	input	Input transmission block statistics distribution info
+DC	output	Output transmission block statistics distribution info

Printer Terminal Model Record

Table G-22 describes the fields in a printer terminal model record.

Table G-22. Printer Terminal Model Record

Offset	Field Name	Description
+0	ptm	Printer terminal model
+2	nptm	Pointer to next printer model
+6	old_pma_usage_count	Count of devices using old printer model attributes
+A	chapma_taskid	CHAPMA command processor task_id
+E	apec	Auto page-eject channel
+F	bofc	Bottom-of-form channel
+10	mvl	Maximum entries in VFU
+11	cdc_defined_printer_model	Flag for Control Data-defined printer model
+11:1	fl	Fold line
+11:2	vtf	VFU top form
+11:3	micro_substitution	Micro substitution to be done for FPP
+12	fpp	File prefix procedure name identifier
+14	fpp_status	File prefix procedure load status
+15	ffs	Form-feed sequence
+1D	fps	File prefix sequence
+3D	fss	File suffix sequence
+5D	nss	No space sequence
+65	sss	Single space sequence
+6D	els	Eight LPI sequence
+75	sls	Six LPI sequence
+7E	Connection pro name id	Connection name ID
+80	cs	Connection sequence
+A0	num_subst_entries	Number of character substitution values
+A2	subst_values	Character substitution values
+AE	ssd	Single space/no space delay count
+7E	ffd	Form-feed (skip channel 1) delay count
+B0	cssp	Channel skip sequences record pointer

Task and Queue Control Blocks H

The task control block (TCB) describes a task to the Executive. The Executive manages TCBs using the queue control block (QCB). Both structures are defined in this appendix.

Information from a TCB found in a CDCNET dump file can be displayed using the DISTCB Dump Analyzer subcommand. Information from a QCB and its buffers can also be displayed using the DISTCB Dump Analyzer subcommand.

Table H-1 summarizes the fields in a TCB. All offsets are expressed in hexadecimal.

Table H-1. Task Control Block

Offset	Field Name	Description
+0	next_task	Chain to next task_ptr
+4	id	This field must be TCB
+8	stsiz	The size of the current task segment
+C	chldq	A pointer to the next sibling task
+10	adult	A pointer to the parent task
+14	child	A pointer to the next child task
+18	stack	Address of the current stack segment
+1C	state_fill	Not used
+1C:4	state	The current state of this task: 0 Rigor Mortis 1 Ready 2 Running 3 Primitive Failure 4 Wait 5 Wait for any Message 6 Wait for Express Message 7 Suspend 8 Wait for any message or wakeup 9 .. 15 Not used (INVALID)
+1D	transition_fill	Not used
+1D:3	trans	Transition that entered this state
+1E	tran	Count of transitions to date
+42	slices	Count of time slice overruns to date

(Continued)

Table H-1. Task Control Block *(Continued)*

Offset	Field Name	Description
+44	flag_fill_1	Upper five bits not used
+44:5	preempted	Flag: task has been preempted; registers all saved (else only A6 and D7)
+44:6	hold	Flag: used by timer task to deflect timer requests into normal queue
+44:7	wku	Flag: wakeup pending if set
+45	flag_fill_2	Not used
+46	express	The express ITM QCB
+56	normal	The normal ITM QCB
+66	preempt_permit	If zero, this task is not preemptible; otherwise, preemptible
+68	cpriority	Nominal priority
+6A	priority	Actual priority
+6C	d_registers	Only register D7 normally valid
+8C	a_registers	Only register A6 normally valid
+A8	usp	User stack pointer
+AC	sr	Status register
+AE	pc	Pointer to program counter
+B2	tcbfrb	Pointer to task failure recovery block
+B6	tcb_epa	Pointer to task entry point
+BA	tcb_space	Amount of unused space in reserved stack area
+BE	tcbmhp	Pointer to module header
+CA	age	Age of task in dispatch queue
+CC	tcb_mem_own_id	Memory/buffer owner identifier
+CE	tcb_itm_length	Length in words of last directly copied task

Table H-2 summarizes the fields in a QCB. All offsets are expressed in hexadecimal.

Table H-2. Queue Control Block

Offset	Field Name	Description
+0	length	Number of items currently queued
+2	count	Running number of items that have been queued to this QCB
+4	qnext	Pointer to next item in queue
+8	qlast	Pointer to last item in queue
+C	qcharacters	Number of characters in queue

Stack Frames I

This appendix describes the stack frame structure, which is used to chain procedure calls for CDCNET tasks.

Each DI task that has been started has a task control block (TCB) that identifies (among other things) the stack starting address and length.

A stack's starting address and length can be found using the Dump Analyzer DISTCB subcommand. For example, the following subcommand displays information from the TCB at address 278B6C(16):

```
DISTCB TI=278B6C DO=FULL
```

Output from this subcommand is formatted as follows:

```
TASK CONTROL BLOCK DISPLAY

TCB ADDRESS                                                  278B6C(16)
TASK NAME      ASYNCTIP_MODULE
NEXT_TASK    { CHAIN TO NEXT TASK POINTER }                       0(16)
ID           { TCB IDENTIFICATION }                                'TCB
STSIZ        { SIZE OF CURRENT STACK SEGMENT }                  4A0(16)
CHLDQ        { TASK POINTER OF NEXT SIBLING }                2290CA(16)
ADULT        { TASK POINTER OF PARENT }                      106072(16)
CHILD        { TASK POINTER OF CHILD }                            0(16)
STACK        { ADDRESS OF CURRENT STACK SEGMENT }             23CF0(16)
STATE        { CURRENT STATE }                     WAIT FOR ANY MESSAGE
PRIORITY     { ACTUAL PRIORITY }                                      0
EXPRESS ITM  { NUMBER EXPRESS INTER_TASK MSGS QUEUED }            0(10)
NORMAL ITM   { NUMBER NORMAL INTER_TASK MSGS QUEUED }             0(10)
SR           { STATUS REGISTER }
               TRACE MODE                NO
               M68000 MODE               USER
               INTERRUPT MASK            000
               RESULT EXTENDED           YES
               RESULT NEGATIVE           NO
               RESULT ZERO               NO
               OVERFLOW                  NO
               CARRY                     NO
PC           { PROGRAM COUNTER }                              1016E(16)
A6           { STACK POINTER }                                2404A(16)
A7           { STACK POINTER }                                2404A(16)
               MODULE NAME               EXEC_PMM
               OFFSET IN CODE SECTION        142(16)
OWNER_ID     { ID GENERATED FOR MEMORY OWNERSHIP }             F0BF(16)
```

The fields named STSIZ and STACK give you the stack length and stack first byte address, respectively. You can use these two values to display stack memory. Use the DISM subcommand, from the stack starting address for stack number of bytes, as follows:

```
DISM A=23CF0 RC=4A0
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS      23CF0

HEX ADDR              HEXADECIMAL DATA                    ASCII DATA

    23CF0  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D00  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D10  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D20  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D30  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D40  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D50  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D60  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D70  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D80  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23D90  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23DA0  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23DB0  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23DC0  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23DD0  4F56 4552 464C 4F57 4F56 4552 464C 4F57   OVERFLOWOVERFLOW
    23DE0  7374 6163 6B7E 0000 7374 6163 6B7E 0001   stack~  stack~
    23DF0  7374 6163 6B7E 0002 7374 6163 6B7E 0003   stack~  stack~
                                                          .


    24000  0017 0021 E8EE FFFF FFFF 0018 12D0 001E      !hn        P
    24010  8E80 0000 002C 0018 12D0 0028 9674 0002         ,   P ( t
    24020  4040 001E 8D50 0002 4058 0027 0002 40E4   @@   P  @X '   @d
    24030  001A 83C2 0000 0021 E89C 412A 0028 0002      B    !h A* (
    24040  4062 0027 E064 0021 E89C 0002 406E 001F   @b ' !h   @n
    24050  BD40 0000 0010 0002 4114 001F E4EA 001E   =@       A    dj
    24060  A3FC 0027 8456 0027 8456 0001 0002 0002   #| ' V ' V
    24070  417C 0027 E99A 0010 7980 0002 4170 0002   A| 'ı    y0  Ap
    24080  4114 0027 8456 0027 83DA 0010 793C 0010   A   ' V ' Z  y<
    24090  7300 6163 6B7E 0000 0000 0021 E850 0057   s ack~      !hP W
    240A0  7374 6163 6B7E 0058 7374 6163 6B7E 0059   stack~ Xstack~ Y
    240B0  7374 0021 F008 005A 7374 6163 6B7E 005B   st !p  Zstack~ [
    240C0  7374 6163 6B7E 005C 7374 6163 6B7E 005D   stack~ \stack~ ]
    240D0  7374 6163 0007 0027 7D9E 0000 0000 000C   stac   '}
    240E0  0011 7440 0027 DFB4 0002 417C 0027 DC16     t@ '_4  A| '\
    240F0  0002 417C 007E 0062 0000 0002 007E 0063     A| ~ b      ~ c
    24100  0000 0000 0005 0002 0001 0000 007E 0065                 ~ e
    24110  0000 0000 0107 0027 7D9E 0080 0011 7440            '}     t@
    24120  0000 0000 6B7E 0068 7374 0028 9674 0000        k~ hst ( t
    24130  0000 0011 7440 0000 2580 0000 0000 0000        t@  %
    24140  0001 0000 0000 0000 0000 0000 0000 0000
    24150  0000 001F FC8A 0022 686A 0022 686A 0021         |  "hj "hj !
    24160  E89C 0021 F008 0080 0028 9462 0028 943C   h  !p     ( b ( <
    24170  0001 0000 0002 0003 7374 6163 0002 4180             stac  A
    24180  0010 E86E 0000 0000 0000 0000 0000 0000     hn
```

The lowest-addressed 240 bytes of the stack are the reserved stack area. This area is preset with the pattern OVERFLOW. The rest of the stack area is preset with repetitions of the hexadecimal value 7374 6163 6B7E nnnn, where nnnn increments from 0000 to whatever value is necessary in order to fill the stack area. This hexadecimal number evaluates to the ASCII string stack~aa, where aa is the two-character ASCII string associated with hexadecimal value nnnn.

Program calls are chained using stack frames, which are written into the stack from the stack's high address toward its low address. The procedure call associated with the lowest-addressed stack frame is the only one that may execute, although all stack frames from there to the high address of the stack area may be considered active. When a program exits, its stack frame becomes inactive and the procedure call of the previous stack frame executes. The stack memory area associated with inactive calls is not reset to the stack¯aa pattern.

Stacks should be long enough so that the reserved stack area is not overwritten. The Dump Analyzer VALSA subcommand identifies tasks whose reserved stack areas have been overwritten.

A stack frame may contain the following:

- A pointer to the previous stack frame (A6)
- An area for local variables and/or compiler temporary storage
- Any parameters being passed on the next call
- A procedure return address (RA)

Figure I-1 shows the relative locations of information in a stack.

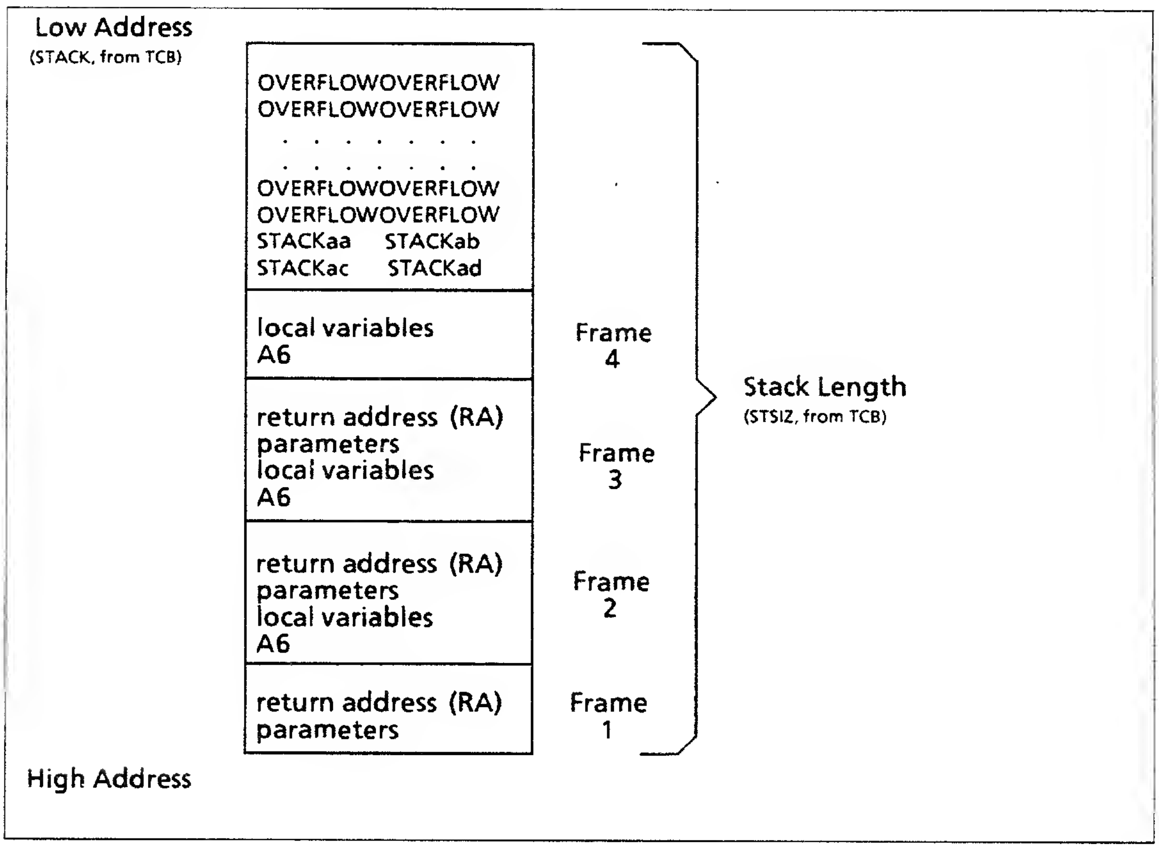


Figure I-1. Stack Area Structure

The TCB maintains two pointers to locations within the stack:

- A pointer to the A6 value in the lowest-addressed stack frame
- A pointer to the lowest active address in the stack. This is known as the user stack point (USP), or A7

These two pointers are contiguous in the TCB. They may be displayed with the Dump Analyzer DISM subcommand. Use the address of the TCB for the desired task on the address parameter, a byte offset (BO) of 0A0(16), and a byte count (BC) of 8. For the example in progress, use the following subcommand:

```
DISM A=278b6c BO=0A0 BC=8
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS    278C0C

HEX ADDR            HEXADECIMAL DATA            ASCII DATA

  278C0C  0002 404A 0002 404A                    @J  @J
```

where the first four bytes contain the A6 value of the lowest-addressed stack frame in the stack and the second four bytes are the USP (or A7).

The stack frames can be displayed individually using the Dump Analyzer DISPLAY_CALL subcommand, with SF=TRUE. For example:

```
DISC TI=278B6C SF=TRUE
```

Output from this subcommand is formatted as follows:

```
TCB ADDRESS          278B6C(16)

    TRACEBACK FROM MODULE   EXEC_PMM + 142(16)
     NEAREST PRECEDING ENTRY POINT   CALL_SURE_BG

  STACK FRAME STARTING AT     2404A(16) AND ENDING AT     2406D(16) .

         0 1  2 3  4 5  6 7  8 9  A B  C D  E F
         ---------------------------------------
   2404A 0002 406E 001F BD40 0000 0010 0002 4114     @n  =@      A
   2405A 001F E4EA 001E A3FC 0027 8456 0027 8456     dJ  #| ' V ' V
   2406A 0001 0002

   CALLED FROM  - MODULE  LINE_CONTROL_BOUND + 4C02(16)
    NEAREST PRECEDING ENTRY POINT   TS_DEBUG_GET_MSG

  STACK FRAME STARTING AT     2406E(16) AND ENDING AT     2417B(16)

         0 1  2 3  4 5  6 7  8 9  A B  C D  E F
         ---------------------------------------
   2406E 0002 417C 0027 E99A 0010 79B0 0002 4170     A| '1   y0  Ap
   2407E 0002 4114 0027 8456 0027 83DA 0010 793C     A   ' V ' Z  y<
   2408E 0010 7300 6163 6B7E 00D0 0000 0021 E850     s ack~      !hP
   2409E 0057 7374 6163 6B7E 0058 7374 6163 6B7E    Wstack~ Xstack~
   240AE 0059 7374 0021 F008 005A 7374 6163 6B7E    Yst !p  Zstack~
   240BE 005B 7374 6163 6B7E 005C 7374 6163 6B7E    [stack~ \stack~
   240CE 005D 7374 6163 0007 0027 7D9E 0000 0000    ]stac    '}
   240DE 000C 0011 7440 0027 DFB4 0002 417C 0027       t@ '_4  A| '
   240EE DC16 0002 417C 007E 0062 0000 0002 007E   \   A| ~ b       ~
   240FE 0063 0000 0000 0005 0002 0001 0000 007E    c                ~
   2410E 0065 0000 0000 0107 0027 7D9E 0080 0011    e          '}
   2411E 7440 0000 0000 6B7E 0068 7374 0028 9674   t@     k~ hst ( t
   2412E 0000 0000 0011 7440 0000 2580 0000 0000          t@  %
   2413E 0000 0001 0000 0000 0000 0000 0000 0000
   2414E 0000 0000 001F FC8A 0022 686A 0022 686A          |  "hJ "hJ
   2415E 0021 E89C 0021 F008 0080 0028 9462 0028     !h  !p    ( b (
   2416E 943C 0001 0000 0002 0003 7374 6163         <     .   stac

   CALLED FROM  - MODULE  ASYNCTIP_MODULE + 16C4(16)

  STACK FRAME STARTING AT     2417C(16) AND ENDING AT     2417F(16)

         0 1  2 3  4 5  6 7  8 9  A B  C D  E F
         ---------------------------------------
   2417C 0002 4180                                   A

   CALLED FROM  - MODULE  OLL_PROGRAM_INTERFACE_PROCS + 1772(18)
    NEAREST PRECEDING ENTRY POINT   START_NAMED_TASK_AND_PROCEED
```

Dump Analyzer Error Messages J

This appendix describes the Dump Analyzer error messages. Command parser errors encountered while running the Dump Analyzer under NOS are documented in the NOS Version 2 Operations Handbook; parser errors encountered under NOS/VE are documented in the NOS/VE Diagnostic Messages manual.

All of the following error messages have a product code of DA. If you are using the Dump Analyzer in FULL message mode, the product code and a unique message number appear in the message text immediately following the severity level. The error messages in this appendix are documented in the FULL message mode.

If you are using the Dump Analyzer under NOS, FULL message mode is the default. Under NOS/VE, BRIEF message mode is the default. With BRIEF message mode, neither the product code nor the message number are displayed. Because the messages in this appendix are sorted by message number, use the FULL message mode when identifying messages. See the NOS/VE Commands and Functions manual for a description of the SET_MESSAGE_MODE command, which lets you change from one message mode to the other under NOS/VE.

Message severity levels are ordered as follows, from the most severe to the least. To the right of each severity level are the numbers of the messages classified at that severity level.

Catastrophic	9, 12, 26, 77, 78
Fatal	5-7, 10, 19, 20, 23, 27, 29, 31, 32, 38-46, 48, 49, 51, 128-131
Warning	0, 1, 4, 21, 24, 25, 30, 33-35, 37, 47, 52, 79-82, 84-86, 89-95, 97-104, 106, 110-113, 118-124, 127, 132
Error	2, 3, 8, 11, 13-18, 22, 28, 36, 88, 96, 105, 107-109, 114-117, 125, 126, 133-140
Informative	83, 87
Reserved	50, 53-76

NOTE

Error messages denoted with asterisks (*) are only applicable to CDCNET version 1.0, as these errors are detected by a standard parser in later releases. Error messages denoted with double asterisks (**) occur only during the integration (build) phase.

Following are the messages and their descriptions. They are sorted by message number.

--WARNING DA 0-- The dump file value for [text] is invalid.

Description: The contents of the dump file corresponding to the specified structure are out of range, preventing further processing of the command.

User Action: Be aware that other commands using the specified structure may result in the same error.

--WARNING DA 1-- Not all memory for the [text] at address [text] through [text] is in the dump file.

Description: The specified structure at the specified memory locations is not completely contained in the dump file.

User Action: A DISPLAY_MEMORY of the specified addresses will show the available parts of the structure.

--ERROR DA 2*-- Byte offset value specified was too big.

Description: The byte_offset parameter of the previous command was greater than the maximum memory address.

User Action: Reenter the command with a byte offset parameter that is within range. See the Dump Analyzer ERS for the proper range of the byte_offset parameter.

--ERROR DA 3*-- Incorrect parameter value specified.

Description: The value of the parameter specified with the previous Dump Analyzer command is out of range. This message will no longer be used in R1.2 and beyond.

User Action: Reenter the command with the proper parameter value. See to the Dump Analyzer ERS for correct parameter ranges.

--WARNING DA 4-- TCB tree list limit exceeded.

Description: The internal limit of task control blocks was exceeded while building the TCB tree list, preventing successful construction of the list. This error suggests corruption of the dump file or misbehavior of the task scheduler as the number of TCBs exceeds the reasonable limit supported by the Dump Analyzer.

User Action: Examine the output from DISTCB ALL which may show where invalid task control blocks begin.

--FATAL DA 5-- Internal error - Data length exceeds 48 bits.

Description: Internal error in Dump Analyzer field packing process.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--FATAL DA 6-- Internal error - Data length is not a multiple of 8 bits.

Description: Internal error in Dump Analyzer field packing process.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--FATAL DA 7-- Internal error - Byte offset in field is out of range.

Description: Internal error in Dump Analyzer field packing process.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--ERROR DA 8-- Expected command, found [text].

Description: The string specified is not a valid Dump Analyzer command.

User Action: Enter a legal Dump Analyzer command.

--CATASTROPHIC DA 9-- The auto dump table contains only one entry.

Description: The auto dump table is used by the Dump Analyzer to process a DI dump. It consists of a number of entries with each entry specifying the address and length of a DI process structure. The first entry defines the rest of the ADT itself - if this is the only entry, no other DI structure can be read and processed.

User Action: Since this dump cannot be processed by the Dump Analyzer, it must be analyzed manually or another dump file must be created and used.

--FATAL DA 10-- Sufficient storage not available.

Description: The Dump Analyzer allocates storage during execution in order to build module lists, TCB tree lists, etc. so various commands can be completed. This error message indicates the Dump Analyzer job has exceeded its allowed memory usage. Any further subcommands requiring memory allocation will not complete.

User Action: Rerun the Dump Analyzer job with more memory specified. If this error continues to occur when executing the same command, it may indicate the structure being processed contains corrupted values. Further examination of the structure using the DISPLAY_MEMORY command should locate the problem.

--ERROR DA 11-- Command expected but not found.

Description: The Dump Analyzer did not recognize the latest input as a a valid command.

User Action: Enter a valid Dump Analyzer command.

--CATASTROPHIC DA 12-- Dump file [text] not found or contains no data.

Description: The specified dump file was not found or contains no data. The Dump Analyzer on NOS must have the dump file attached as a local file. This is not a restriction on on NOS/VE.

User Action: Recheck accessibility and size of the dump file and re-invoke the Dump Analyzer.

--ERROR DA 13*-- Address value specified is too big.

Description: The address value entered exceeded the maximum Dl memory address.

User Action: See the Dump Analyzer ERS for the maximum address value permissible and reenter the command with an address value that is within range.

--ERROR DA 14-- The sum of the address and byte_offset parameters is too big.

Description: The sum of the address and byte_offset values exceeded the maximum Dl memory address.

User Action: See the Dump Analyzer ERS for the maximum permitted value. Reenter the command, insuring the sum of the address and byte-offset parameters does not exceed this maximum.

--ERROR DA 15*-- A negative value was specified for the address value.

Description: The address value specified was less than zero.

User Action: Reenter the command with an address value that is within range. Refer to the Dump Analyzer ERS for the correct range.

--ERROR DA 16*-- A negative value was specified for the byte_offset parameter.

Description: The byte_offset value specified was less than zero.

User Action: Reenter the command with a byte_offset value that is within range. Refer to the Dump Analyzer ERS for the range allowed for the byte_offset parameter.

--ERROR DA 17*-- The value specified for byte_count must be greater than zero.

Description: The byte_count value entered was less than or equal to zero.

User Action: Refer to the Dump Analyzer ERS for the range allowed for the byte_count parameter. Reenter the command with a byte_count parameter that is within range.

--ERROR DA 18*-- The value specified for repeat_count must be greater than zero.

Description: The repeat_count value entered was less than or equal to zero.

User Action: Refer to the Dump Analyzer ERS for the range allowed for the repeat_count parameter. Reenter the command with a repeat_count parameter that is within range.

--FATAL DA 19-- The number of bytes read does not match auto dump table info.

Description: The auto dump table consists of entries specifying the address and length of a Dl structure and is used to read the dump file. The latest read performed by the Dump Analyzer did not match the length specified in the ADT. This error suggests a corrupted dump file.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--FATAL DA 20-- Internal error - invalid bit displacement specified during alignment of dump file data.

Description: Internal error - the bit displacement of the requested field is not on an 8-bit (byte) boundary.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--WARNING DA 21-- Specified stack frame does not exist.

Description: The stack frame associated with the structure being displayed does not exist.

User Action: Be aware that the command may not have completed. If this error occurs after a DISC command specifying a parameter, reenter DISC with START=1 and COUNT=ALL to display all available stack frames.

--ERROR DA 22*-- The COUNT parameter must be a positive integer or ALL.

Description: The value specified for the COUNT parameter was not within the stated range.

User Action: Reenter the command with a COUNT parameter that is within range. Refer to the Dump Analyzer ERS for the correct range of the COUNT parameter.

--FATAL DA 23-- Internal error - Field crossed a byte boundary.

Description: Internal error in aligning data in the Dump Analyzer field packing process.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--WARNING DA 24-- Invalid child task pointer [text] was encountered while building the TCB tree list. The chain cannot be followed - some TCBs may not be accessible to future commands.

Description: While building the TCB tree list for use by the DISTCB command, a corrupted TCB was found at the specified address. Any further TCBs along that branch of the tree cannot be processed, but the rest of the TCBs are valid.

User Action: A DISPLAY_MEMORY of the specified address will show what the corrupted TCB looks like and may yield clues as to why it is corrupted.

--WARNING DA 25-- The dump file value for [text] is invalid. Some commands may be affected.

Description: The specified structure is invalid. Commands referencing this structure may give incomplete results.

User Action: Be aware of this fact when using commands involving the specified structure.

--CATASTROPHIC DA 26-- The dump file value for [text] is unusable. No further processing possible.

Description: The specified structure is invalid, causing termination of the Dump Analyzer.

User Action: This dump cannot be processed by ANACD. Examine dump manually.

--FATAL DA 27-- The well_known ram locations (including module data pointers) are not available.

Description: Key portions of the MPB_RAM table are not available in this dump, including the module data pointers needed to build the module list.

User Action: Avoid use of commands requiring a module name as a parameter.

--ERROR DA 28*-- The START parameter value must be a positive integer not exceeding 4095.

Description: The value entered for the START parameter was not within the range 1 .. 4095.

User Action: Reenter the command with a START parameter that is within range.

--FATAL DA 29**-- There is no table length entry for record [text].

Description: Build error - the specified record does not have a corresponding table length entry in the module DADTL. The build will not complete.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--WARNING DA 30-- The data at address [text] is not a valid task control block.

Description: The task control block header at the specified address has been corrupted.

User Action: A DISPLAY_MEMORY of the specified address may yield relevant information as some TCB fields may be intact.

--FATAL DA 31-- Internal error - Number of bytes exceeds result parameter size.

Description: Internal error - the specified result parameter is too small to contain the requested number of bytes.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--FATAL DA 32-- Internal error - PACK_STRING_FIELD byte number exceeds buffer size.

Description: Internal error - the highest element to be accessed in the input array exceeds the upper bounds of the array.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--WARNING DA 33-- Entry point [text] at address [text] is not in any section of module [text].

Description: The specified entry point is not contained in the specified module.

User Action: A DISPLAY_MEMORY of the area surrounding the specified address should reveal which module contains the specified entry point. The DISMM command may show more information about the specified module.

--WARNING DA 34-- Invalid dump file value for [text] - entry point data is not available.

Description: The specified DI structure is invalid, preventing availability of entry point data.

User Action: Avoid use of entry point parameters on subsequent commands.

--WARNING DA 35-- Invalid dump file value for [text] - module/section data is not available.

Description: The specified DI structure is invalid, preventing construction of the module list. Commands requiring a module list will not complete.

User Action: Be aware that commands requiring a module list may not complete successfully.

--ERROR DA 36*-- The value specified for display_option must be E, S, or F.

Description: The DISPLAY_OPTION parameter entered was not ENTRY, SECTION, or FULL (or one of their letter abbreviations).

User Action: Refer to the explanation of the DISPLAY_MEMORY_MAP command in the Dump Analyzer ERS for the meaning of the DISPLAY_OPTION parameter. Re-enter the command with a valid DISPLAY_OPTION parameter.

--WARNING DA 37-- A task has not been selected as the current task. Command cannot be completed.

Description: Several Dump Analyzer commands default to the current task in the absence of a TASK_IDENTIFIER parameter. The current task is either the task that was running at the time the dump was taken or a task specified with the SELECT_TASK command. In this case, no task was running when the dump was taken and no specific task has been selected as the current task.

User Action: Use the SELECT_TASK command to specify a current task and reenter the previous command, or reenter the previous command specifying a task by using the TASK_IDENTIFIER parameter.

--FATAL DA 38**-- The file named [text] is not local to this job.

Description: The specified object library file required for the GENFD build utility cannot be found.

User Action: Make the specified file accessible and rerun the job.

--FATAL DA 39**-- Library file [text] does not contain a library header record.

Description: Build error - The file specified as input to the build utility, GENFD, is not recognized as an object library.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 40**-- The library file contains no symbol table records for module [text].

Description: Build error - the object library segment for the specified module is corrupted.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 41**-- The library file does not contain module [text].

Description: Build error - the specified module is not in the object library.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 42**-- The library file contains a format error for module [text].

Description: Build error - the specified module is corrupted in the object library.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 43**-- Module [text] is not of type M680.

Description: Build error - the object library was built incorrectly.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 44**-- The symbol table for module [text] contains a format error.

Description: Build error - the specified module's symbol table is corrupted.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 45-- An unexpected file mark has been found on file [text].

Description: The specified file contains a format error.

User Action: If the specified file is an input to the build utility GENFD, submit a PSR to Control Data. If the specified file is a dump file, no further processing is possible.

--FATAL DA 46**-- There is no symbol table entry for field [text] of record [text].

Description: Build error - the specified field was deleted in the DI software but not in the ANACD module DAMZZCD.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--WARNING DA 47-- Invalid sibling task pointer [text] was encountered while building TCB tree list. The chain cannot be followed - some TCBs may not be accessible to future commands.

Description: While building the TCB tree list for use by the DISTCB command, a corrupted TCB was found at the specified address. Any further TCBs along that branch of the tree cannot be processed, but the rest of the TCBs are valid.

User Action: A DISPLAY_MEMORY of the specified address will show what the corrupted TCB looks like and may yield clues as to why it is corrupted.

--FATAL DA 48**-- There is no symbol table item with symbol number = [text].

Description: Build error - the specified symbol number does not have a symbol table item.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 49**-- There is no symbol table entry for record [text].

Description: Build error - the specified record was deleted in the DI software but not in the ANACD software.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 51-- Index sequential files not supported.

Description: Internal error in file open process.

User Action: Submit a PSR to Control Data with the dump file and the input file directives that caused the error.

--WARNING DA 52-- The value of [text] is [text], which is not within the TCB stack segment.

Description: The specified value of the specified structure is not within the stack frame chain.

User Action: Use the DISM command or the DISMM command specifying the SF parameter to look at the stack segment for further information.

--CATASTROPHIC DA 77-- Dump file length [text] too short to proceed.

Description: The dump file length specified is so small that key dump file structures (such as mpb_ram) are not contained in the dump, making any further processing impossible.

User Action: As this dump cannot be processed by the Dump Analyzer, manual analysis is required. If there is another dump for the same condition, it should be used with the Dump Analyzer.

--CATASTROPHIC DA 78-- Internal error - Auto Dump Table entries exceed the limit of [text]. It is doubtful that this is a valid CDCNET dump.

Description: The number of auto dump table entries exceeds the specified maximum, precluding further processing by the Dump Analyzer.

User Action: The high number of ADT entries indicates corruption of the dump file. Manual analysis may indicate the problem.

--WARNING DA 79-- Auto Dump Table at [text] has been adjusted due to short dump file of length [text].

Description: The size of the dump file was less than the size indicated by the auto dump table. To bring the size indicated by the ADT into agreement with the actual size of the dump, entries were removed from the end of the ADT until the lengths were in agreement.

User Action: Be aware that some information captured by the DI may not be contained in the dump.

--WARNING DA 80-- Command cannot be processed due to absence of SCT in dump file.

Description: The system configuration table is missing from the dump file. Any commands requiring the SCT will not complete.

User Action: Avoid commands requiring the SCT - for example, the DISSCT, DISTCB, and VALSA commands.

--WARNING DA 81-- Loop in TCB chain--entry at [text] points to entry at [text].

Description: A loop has been detected at the specified addresses during construction of the TCB tree list. An alternate method of construction will now be employed.

User Action: Be aware that the command will take significantly longer to complete.

--WARNING DA 82-- TCB chain broken at address [text].

Description: A break was detected at the specified address during construction of the TCB tree list. An alternate method will be employed.

User Action: Be aware that the command will take significantly longer to complete.

--INFORMATIVE DA 83-- Scanning memory for TCB identifier to rebuild TCB chain. This process may take some time.

Description: Due to a flaw in the TCB chain, the TCB tree list will be constructed by scanning memory for the literal !TCB. This method requires significant resource usage.

User Action: Be aware that the command will take significantly longer to complete.

--WARNING DA 84-- The [text] value in the dump file is invalid.

Description: The dump file value for the specified state is out of range. The display field will be filled with asterisks.

User Action: None required.

--WARNING DA 85-- Integer to string conversion error--value cannot be displayed.

Description: An error occurred during conversion of an integer value to a displayable string. The field will be filled with asterisks.

User Action: None required.

--WARNING DA 86-- Dump Analyzer version [text] does not match dump file compiled version [text] recorded in SYSTEM_DATA. Some display fields may contain erroneous values. If possible, use a Dump Analyzer of the same version as the compiled dump file.

Description: The specified build level of the Dump Analyzer differs from the specified build level of the DI software. Since the Dump Analyzer uses DI software tables to access and display the corresponding DI structures, a version mismatch could result in erroneous display of DI structures.

User Action: Re-invoke the Dump Analyzer, specifying a version matching the dump file compiled version. If a matching version does not exist, use a Dump Analyzer version as close as possible to the dump file compiled version.

--INFORMATIVE DA 87-- No task was running at time of dump.

Description: There was no task active at the time the dump was taken.

User Action: User may wish to specify a current task for further processing by invoking the SELECT_TASK command.

--ERROR DA 88-- ADDRESS parameter must be a DI memory address or entry point name.

Description: The ADDRESS parameter specified on the last command was not a valid address or DI entry point name.

User Action: Reenter the command using a valid ADDRESS parameter. Refer to the Dump Analyzer ERS for the correct range of this parameter.

--WARNING DA 89-- Entry point [text] cannot be found in dump file.

Description: The specified entry point was not contained in the dump file's module list.

User Action: Entering the DISPLAY_MEMORY_MAP command will yield a list of all the entry points contained in the dump file.

--WARNING DA 90-- The CDCNET DI software compiled version recorded in SYSTEM_DATA cannot be verified.

Description: The dump file being processed does not contain the SYSTEM_DATA table, which contains the DI software compiled version.

User Action: If the dump file loaded version seems valid and some displays appear erroneous, a Dump Analyzer version matching the dump file loaded version should be used with this dump.

--WARNING DA 91-- Module [text] cannot be found in dump file.

Description: The specified module could not be found in the dump file's module list.

User Action: Invoking the DISPLAY_MEMORY_MAP command will yield a list of all the module names contained in this dump file.

--WARNING DA 92-- TCB at [text] pointed to by running task pointer is invalid.

Description: The task control block at the specified address is corrupted. Since this is the running task TCB, there is now no current task.

User Action: If a current task is desired, it may be specified with the SELECT_TASK command. A DISPLAY_MEMORY of specified address may yield some useful information about the task running at the time of the dump.

--WARNING DA 93-- Dump file value for running task is corrupted: new task must be specified.

Description: The SELECT_TASK command was entered with no TI parameter, and the dump file value for the task running when the dump was taken is invalid.

User Action: Reenter the SELECT_TASK command with a TASK_IDENTIFIER specified.

--WARNING DA 94-- No task was running at time of dump: new task address must be specified.

Description: The SELECT_TASK command was entered with no TASK_IDENTIFIER specified, and no task was running when the dump was taken.

User Action: Reenter the SELECT_TASK command with a TASK_IDENTIFIER specified.

--WARNING DA 95-- Bad link at [text] found in list chain.

Description: The address specified was a bad link pointer discovered when processing a linked list. No further processing is possible.

User Action: Use the DISM command to examine the area surrounding the specified address. This may yield further information.

--ERROR DA 96-- Command [text] not available in this version of the Dump Analyser.

Description: The specified command is not available in this release.

User Action: Use other commands to find the desired information.

--WARNING DA 97-- Buffer length not available in system configuration table-- using buffer size of [text] bytes.

Description: The value for a data buffer's length is not available in the dump file, so the specified length will be used in generating the buffer display.

User Action: If the specified buffer size is known to be incorrect, the known size can be used with the DISPLAY_MEMORY command to view the contents of the buffer.

--WARNING DA 98-- Module header pointer in task control block is invalid.

Description: The pointer to the structure containing the name of the task being processed is corrupted. The task name will be found using an alternate method.

User Action: None required.

--WARNING DA 99-- Line name [text] not found in dump file.

Description: The specified line name is not contained in the dump being analyzed.

User Action: Reenter the command with a different line name or the keyword ALL to see which line names are available in the dump.

--WARNING DA 100-- Header for linked list entry at [text] is too large or linked list entry address too low.

Description: Either the header size specified is too large or the list item address specified is too low. When the header is subtracted from the list item address the result is less than 1 (the minimum DI memory address).

User Action: Reenter the command with different values for either the header size parameter or the linked list starting address. The subcommand may have generated some output before terminating - check this if you think your parameter values were correct.

--WARNING DA 101-- Address of link_offset plus linked list entry at [text] exceeds the maximum DI memory address.

Description: Either the link_offset or the specified list item address is too high. When the link_offset is added to the list item address the result is greater than the maximum DI memory address.

User Action: Reenter the command with different values for either the link_offset parameter or the linked list starting address. The subcommand may have generated some output before terminating - check this if you think your parameter values were correct.

--WARNING DA 102-- Stack extension allocated. Previous stack segment allocated at DI address [text] will not be processed.

Description: A link to a previous stack segment has been detected, indicating that a stack extension has occurred. There may be information of interest in the previous stack frame(s).

User Action: Use DISPLAY_MEMORY to look at previous stack segment at specified address if desired.

--WARNING DA 103-- Output terminated by a user break sequence.

Description: A user break 2 has been detected during the generation of subcommand display output. This sequence has been interpreted as a request to terminate the output generation from the current subcommand. This message appears in the NOS version of ANACD only.

User Action: Continue to enter subcommands at the prompt. Note this message may not appear if the generation of output was already complete before the break sequence was detected. In this case any output waiting to be displayed at the terminal will be discarded so the effect will be the same. If you desire to terminate the ANACD session entirely enter the QUIT subcommand or three (3) user break 2 sequences in succession. Consult your NOS operating system manual for more information on user break sequences. If you are running under NOS/VE this message will be replaced by a standard system message and user breaks will be handled entirely by the operating system in the standard way for NOS/VE utilities. Consult the SCL NOS/VE System Interface manual for further information.

--WARNING DA 104-- File name [text] truncated to [text].

Description: On the NOS operating system file names are limited to 7 characters. The parser for the subcommands will allow the full 31 characters but only the first seven will be used to identify the file on NOS. This warning only refers to files on subcommands which are normally output display files.

User Action: Make a note of the shortened file name. If the short name is the same as one already in use data will be appended as a NOS record. The execution of the subcommand will not be affected.

--ERROR DA 105-- First seven characters ([text]), of specified file name does not constitute a valid NOS file name.

Description: On the NOS operating system file names can only contain alphabetic or numeric characters but on NOS/VE the special characters ($ # @ _) are also allowed. On NOS, file names specified as subcommand parameters are truncated to 7 characters and if any of these characters are in the set of 4 specials above then this message is issued and the subcommand must be rejected.

User Action: Reenter the subcommand with a valid NOS file name.

--WARNING DA 106-- There are more than [text] TCBs in this dump. Only this limited number will be displayed.

Description: The internal limit of task control blocks was exceeded while building the TCB tree list, preventing successful construction of the list. This error suggests corruption of the dump file or misbehavior of the task scheduler as the number of TCBs exceeds the reasonable limit supported by the Dump Analyzer.

User Action: Examine the output from DISTCB ALL which may show where invalid task control blocks begin.

--ERROR DA 107-- Attempting to generate display onto file [text], which is the same as the dump file.

Description: The name of a file specified by an output parameter on either the main ANACD call or a subcommand, is the same as the name of the specified dump file. The output file will not be accepted in order to safeguard the user from overwriting the dump file.

User Action: Choose another name for the output file and reenter command.

--ERROR DA 108-- Output file [text] not writable.

Description: The output file specified on the main ANACD call or the current subcommand was found to be read or execute only and therefore could not be opened for write mode.

User Action: Check your access mode permissions for the specified file or choose another file name and reenter command.

--ERROR DA 109-- TASK IDENTIFIER parameter must be a DI memory address or a task name.

Description: A task is identified by its TCB address or the name of the module that was scheduled as a task. Unscheduled modules are not task names.

User Action: Check the name or address of this task or TCB and reenter command.

--WARNING DA 110-- Task name [text] is contained in more than one TCB - no task selection made.

Description: The SELECT_TASK subcommand allocates a single task or TCB to be the default selection for many task oriented subcommands. If a module has been scheduled more than once it will be the task name of more than one TCB so that the name alone does not uniquely identify the TCB.

User Action: Use DISTCB with the task name (BRIEF mode will do) to find the TCB addresses of all the tasks with that name. Select the TCB desired to be the default or 'selected' one and reenter the SELT subcommand with the TI parameter equal to the desired TCB address.

--WARNING DA 111-- Device name [text] is not currently supported. Device names must be of the form $LIMn where n is the slot number.

Description: The DISPLAY_HARDWARE_STATUS command only supports display selection by LIM and slot number. Selection of other DEVICE_NAMEs has the same effect as omitting the parameter or selecting DN=ALL; that is, status for all the major boards or all the LIMs is displayed.

User Action: None.

--WARNING DA 112-- The [text] is not [text], which is the range expected. This may be due to invalid data in the SMM BANK STATUS TABLE or a new hardware configuration. Check the rest of this subcommand display for further inconsistencies.

Description: The SMM_BLOCK_SIZE or NUMBER_OF_BANKS used to calculate the address of subsequent SMM boards for the hardware status display are outside the expected ranges. This version expects only 1MB boards with a breakdown into 1MB or 0.5MB block sizes. Other values may be valid for upgraded boards but the current version of ANACD will probably be inadequate in displaying the hardware status of a DI configured with upgraded boards.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--WARNING DA 113-- No configured line control block is associated with this port.

Description: The Configuration Table pointer from the structure PORT_STATUS_TABLE is either NIL or the PORT_OWNER in the same table is not LCM_OWNER. This means no CLCB information is available for this port of a LIM (that is, no line name or protocol). This is normal if the slot is empty or the port has not been assigned to an active LIM.

User Action: Ensure that this is consistent with the rest of the displayed hardware status.

--ERROR DA 114-- The value [text] entered for parameter kind [text] is too long.

Description: The parameter string for this application name or integer kind is too long for a valid name or integer of any kind. This message will only appear on NOS/VE where application value kinds have been defined for some parameters; for exampe, an address parameter which may be a symbolic name or an integer whose default radix is hex instead of standard decimal.

User Action: Use the NOS/VE SCL command DISCI to display the parameter specifications for the subcommand being attempted and re-enter the subcommand with all parameter values having fewer characters than the maximum length of a name or the length of an integer of maximum size represented in its longest form.

--ERROR DA 115-- The value [text] entered for parameter kind [text] is not a valid symbolic name.

Description: An invalid name has been entered for a parameter defined to have an application value kind of SYMBOLIC_ADDRESS. The value should be a valid CYBIL name since it must match an entry point or TCB name. This message will only appear on NOS/VE where application value kinds are defined for some parameters.

User Action: Re-enter subcommand with either an integer address value or a valid CYBIL name for those parameters with kind defined as SYMBOLIC_ADDRESS. Use the SCL command DISCI to determine which parameters are defined as this kind.

--ERROR DA 116-- The value [text] entered for parameter kind [text] may only be numeric. No symbolic interpretation is supplied for this parameter.

Description: This message only appears on NOS/VE. The parameter kind named is an application value kind and limits the parameter to numeric values. A name has been encountered for a parameter of this kind which cannot be interpreted for this subcommand.

User Action: Re-enter the subcommand with parameter values which are integers of the kind specified not symbolic names. Use the SCL command DISCI to determine which parameters are of the specified kind if it is not obvious.

--ERROR DA 117-- The parameter string is too long to add default hex radices to integers.

Description: Integer parameters in ANACD are interpreted as being in the base 16 (hexadecimal), unless an explicit radix is included. This is contrary to the decimal default rule for CCL and SCL parsing, therefore hex radices are added to integers without radices before the parameter string is fully parsed. The addition of these radices (the string '(16)') may make the subcommand line longer than the maximum allowed by CCL or SCL and hence make it too long to parse.

User Action: Attempt to reduce the length of the subcommand entry by taking out excess spaces, using abbreviations wherever possible and converting integers in small radices to larger ones. Shorter file names might also be chosen if any are involved. If the message still appears then submit a PSR to Control Data.

--WARNING DA 118-- Address [text] is not within the ranges of allocatable memory on any board. Only allocatable memory extents have usage headers.

Description: This message is associated with the memory user subcommands and describes a situation where an address is outside of the allocatable memory ranges in a DI where memory usage information is located.

User Action: If the message pertains to an address that is an input parameter, reenter the command with an address that is within range. You can see the valid ranges by displaying memory at the entry point memory_chain_end_points. Note though that the PMM end points are ignored if the SCT shows no bytes in private memory.

--WARNING DA 119-- Memory header address [text] is not present in the dump file. The next memory header address, [text], is beyond the address specified on the request.

Description: The extent containing the address specified on the subcommand request is probably not present in the dump. The first address specified in the message is the start of a nondumped extent, the second address is the start of the next extent that is in the dump.

User Action: Re-enter the subcommand with a different address.

--WARNING DA 120-- No memory found for ownerid [text].

Description: The Dump Analyzer has searched the memory extent areas in this dump and could not find any memory 'owned' by the specified ownerid.

User Action: None required.

--WARNING DA 121-- The specified memory user [text] has no associated memory.

Description: The Dump Analyzer has searched the memory extent areas in this dump and could not find any memory 'owned' by the specified memory user.

User Action: None required.

--WARNING DA 122-- No ownerid found for description [text].

Description: The Dump Analyzer has searched the predefined ownerid array and could not find an ownerid associated with the specified description.

User Action: Check the input description for an exact match with one of the descriptions in the predefined ownerid array.

--WARNING DA 123-- [text] is not a valid owner_id.

Description: The specified value is not within the range of valid owner_ids.

User Action: Refer to the DISPLAY_MEMORY_USERS command description in this document for the range of valid owner_ids.

--WARNING DA 124-- This command may take some time. There are [text] owner_ids for which memory use will be displayed.

Description: Several lists have to be searched to determine all the memory extents owned by all the distinct owner_id's. The exact time taken depends on the machine and its load, but will vary directly with the number of owner_id's quoted.

User Action: Be prepared to wait or terminate the subcommand with a user break 2. There will be a substantial amount of output displayed. Consider terminating and restarting the subcommand with output directed to a mass storage file, if it is currently directed to the terminal.

--ERROR DA 125-- The overlay for command [text] was not found on the overlay file.

Description: The code for the specified command cannot be loaded from the load file. This error is not caused by any user action, but indicates something wrong with the building or installation of the ANACD utility. Only applies to NOS.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error. In the mean time, attempt to use another version of ANACD, as close as possible to the version of your dump file.

--ERROR DA 126-- The A7 pointer cannot be greater than the A6 pointer.

Description: This error can occur while tracing stack frames generated by procedure calls in the D1. A7 is normally equal to A6, or sometimes smaller for partial frames. An A7 greater than A6 would imply less than zero bytes in the frame and hence no further frames can be displayed.

User Action: Inspect the A6 and A7 values shown in a full display of the TCB, to determine which one of them is corrupt.

--WARNING DA 127-- The ANACD_DIRECTIVES parameter will be ignored when DISPLAY_OPTION = BRIEF.

Description: The DISPLAY_MEMORY_USERS subcommand will produce a file of ANACD directives if the AD parameter is specified. However the directives are only generated for extents for which the extent type and extent size are displayed (that is, when display_option = full). The brief display merely summarizes or counts extents and bytes, so there will be no directives generated.

User Action: Re-enter the subcommand with the full display option if the directives are desired.

--FATAL DA 128**-- There are no modules present on the object library.

Description: This condition only occurs during execution of the build tool GENFD on a NOS/VE system. The object library input to GENFD contains no module information and therefore will contain no symbol tables from which to calculate the latest field lengths and offsets.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 129**-- Module is not a recognized load module type.

Description: This condition only occurs during execution of the build tool GENFD on a NOS/VE system. The object library input to GENFD does not contain module information that can be used to calculate the latest field lengths and offsets.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 130**-- Dictionary type is not a module dictionary.

Description: This condition only occurs during execution of the build tool GENFD on a NOS/VE system. The object library input to GENFD contains a type of dictionary which is unsuitable for locating the symbol definitions needed to calculate the latest field lengths and offsets.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--FATAL DA 131**-- Object library version is not supported.

Description: This condition only occurs during execution of the build tool GENFD on a NOS/VE system. The object library input to GENFD is of a version and hence format that this tool cannot interpret.

User Action: Submit a PSR to Control Data with the dump file and the input directives that caused the error.

--WARNING DA 132-- Invalid owner_id of [text] read from [text] address [text].

Description: An owner_id out of the valid range has been read from the dump file, either from an extent header or a TCB. This owner_id cannot be used as a key for further searches.

User Action: Use DISPLAY_MEMORY or DISPLAY_MEMORY_HEADER to investigate further for clues to corruption or initialization problems if the owner_id was read from a memory extent header. Use DISPLAY_TASK_CONTROL_BLOCK if the owner_id was read from a TCB.

--ERROR DA 133-- Node identifier does not match tree root identifier.

Description: The tree_root_identifier, as defined in the root record, does not match the node identifier as defined in the node record for this node of the tree. An error has probably occurred in the building of the tree.

User Action: Investigate the tasks responsible for creating the tree and adding to it.

--ERROR DA 134-- The node value for [text] is invalid.

Description: The value for [text] as read from the dump file appears to be corrupted. The validity or usefulness of this value is questionable.

User Action: Use DISPLAY_MEMORY to investigate further for clues to corruption or initialization problems.

--ERROR DA 135-- The node_control value for [text] is invalid.

Description: The value for [text] as read from the dump file for this particular node control record appears to be corrupted. The validity or usefulness of this value is questionable.

User Action: Use DISPLAY_MEMORY to investigate further for clues to corruption or initialization problems.

--ERROR DA 136-- The specified tree is empty, or possibly not a tree structure.

Description: The tree specified has no branches. The node pointer, as specified in the node control record, is NIL. The value may not have been initialized properly. It is also possible that this is not a tree structure at all.

User Action: Use DISPLAY_MEMORY to investigate further for clues to corruption or initialization problems. Determine that this is indeed a tree structure.

--ERROR DA 137-- The specified key value is invalid.

Description: The key type specified is invalid. Only numeric, pointer, and string type keys are currently supported. Corruption of the key type might be considered as a possible cause.

User Action: Use DISPLAY_MEMORY to investigate further for clues to corruption or initialization problems.

--ERROR DA 138-- The specified log queue type is invalid.

Description: The key type specified is invalid.

User Action: Refer to the CDCNET Commands Reference manual for queue type parameter values that are are currently supported.

--ERROR DA 139-- The specified log queue message buffer pointer is invalid.

Description: The log message buffer pointer, as read from the dump file, is invalid. It might be corrupted.

User Action: Use DISPLAY_MEMORY to look for clues to corruption or initialization problems.

--ERROR DA 140-- The specified log queue message identifier is invalid.

Description: The log message identifier, as read from the dump file, is invalid. It might be corrupted.

User Action: Use DISPLAY_MEMORY to look for clues to corruption or initialization problems.

Index

D

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOLD

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 4760 St. Paul, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

CDCNET Network Operations and Analysis 60461520 J

We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other ______

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up
- ☐ Other ______

What programming languages do you use? ______

How do you like this manual? Answer the questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Do you use this manual frequently?

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply: ☐

Name ______ Company ______

Address ______ Date ______

Phone ______

Please send program listing and output if applicable to your comment.

10260803 31

CONTROL DATA